2001年12月1日発行（毎月1回1日発行）第20巻第12号 昭和57年12月6日第三種郵便物認可

マイコン

わかる！動かせる！
プログラムが組める雑誌

DEC.2001 12

# BASIC Magazine



定価 550Yen

次世代OSの注目機能とインストール&PCアップグレード・テクニックを徹底紹介!!

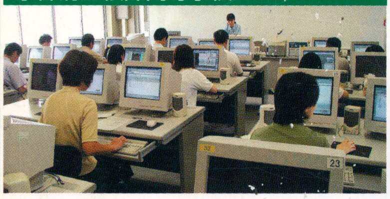


どこが変わった…？操作方法と追加機能

# Windows XP 導入実践講座

## 編集部の学校訪問記

頑張る先生にスポット
京都府立綾部高等学校



Excelを使用した授業
長野県豊科町立豊科北中学校



## プログラミング・テクニック

はじめようMicrosoft ExcelでのBASICプログラミング
**コンピュータ対戦型のパズルゲーム**

DirectXテクノロジーを伝授する! IE+DirectAnimationマルチメディア・プログラミング
**キー・イベントの処理テクニック**

図解！Visual Basicゲーム作成術
**キャラクター・アクション・ゲーム③追跡動作**

携帯電話で動くゲームを作ろう!!
**フォントクラスを使ったiアプリに挑戦**

PS2用「BASIC STUDIO　パワフルゲーム工房」入門
**レースゲーム風デモを作る**

## マンガで学ぼう！パソコンレクチャー
CPUって電子頭脳!?の巻

## ● ロボット／電子工作

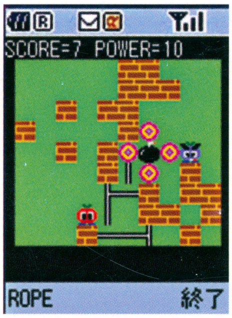
①チャレンジ！レゴ・マインドストーム～センサのちょっと進んだ使いかた
②自作一代記～マイコンでBASICが動く！「AKI-H8マイコンボード」の製作
③テクノ・トイズ・ラボラトリ～ポケットライト・デラックス後編

## 読者が作った作品をドーンと紹介
# 特選投稿プログラムコーナー

| | |
|---|---|
| VisualBasic6.0 | WIND |
| VisualBasic6.0 | 目指せ!! 千切りマスター |
| VisualBasic6.0 | (実用) スケルトン |
| VisualBasic5.0 | T-Racer for BM |
| VisualBasic6.0+DirectX | (超大作) どっち？ |
| VBScript+DirectX | Touch Down |
| VisualC++V6.0 | (超大作) 捕獲屋さん |
| iアプリ | iSCROPE |
| iアプリ | イカクンガンバレfor i-mode |
| PC-9801(N-88BASIC) | メロンヴァー・スイカヴァー |
| Z-1GR/Z-1/FX-890P | TipuraVer4.37 |

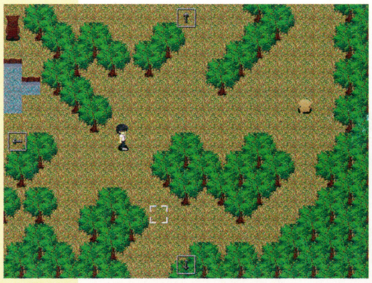
●CMM試聴室

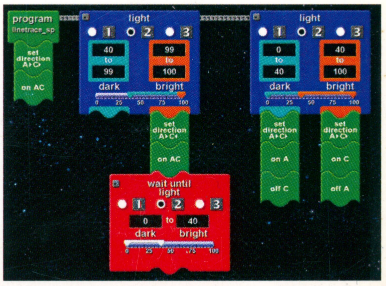
| | |
|---|---|
| MU90B | LAST SUMMER |
| QY70 | Holy Arrow |











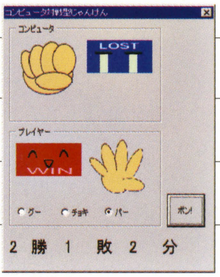
専用ホームページ
http://www.basicmagazine.net/

マイコン BASIC Magazine

# CONTENTS

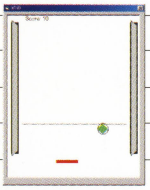
12
DEC.2001

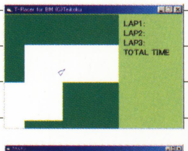


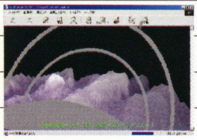


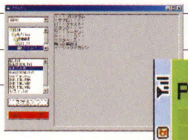

読者が作った作品をドーンと紹介 **特選投稿プログラム・コーナー**

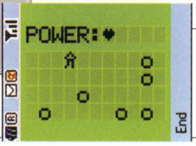















専用ホームページ http://www.basicmagazine.net/

## 特別レポート



## 好評連載　プログラミングから電子工作まで



### 広告さくいん




©マイコンBASICマガジン12月号
第20巻 第12号 2001年11月25日印刷
2001年12月1日発行
定価550円（本体524円）
電波新聞社　2001 Printed in Japan
編集発行人：平山哲雄
東京都品川区東五反田1-11-15
郵便番号141-8715
電話　東京(03)3445-6111（大代表）
振替　（東京）00150-3-51961
印刷：奥村印刷(株)　製本：(株)堅省堂




経済産業省では，毎年，次代を担う青少年の情報処理技術の向上を目的として昭和55年から「全国高校生・専門学校生プログラミング・コンテスト」を実施しています。

22回目となる今年度の表彰式が「平成13年度情報化月間記念式典」の会場で行われました。募集内容は，●プログラミング（高校生の部／専門学校生の部）●コンテンツ（高校生の部／専門学校生の部）となっています。

さっそく，それぞれの入選作品を紹介してみましょう。

# これが 平成13年度全国高校生・専門学校生プログラミング・コンテスト入選作品だ！

## プログラミング（高校生の部）



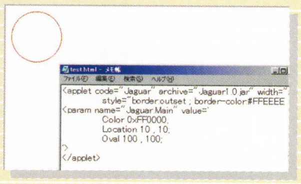

**作品名** Jaguar 1.0

北海道立北海道有朋高等学校
通信制課程　普通科３年
廣瀬貴一

**表彰理由：**

初心者でも簡単に動的なWebページを作成できるようにするサポート・ソフト。高校ではあまり教育が行われていないJAVA言語を用いて実現した点が高く評価された。技術的にもよくできた作品。




### 作者から一言

もともとJavaで，アプリケーションなどを制作していました。簡単なアプレットをWebに使いたいとき，小さなプログラムでもJDKで開発し，コンパイルしなければならず非常に面倒でした。そこで，Webでプログラム的な描画処理をHTMLソース内にスクリプトで記述し，テキストによるインタプリタで対話的に開発するために作成しました。

私がこのプログラミング・コンテストのことを知ったのがちょうど締め切りの1か月ほど前だったので，開発期間が短く，外部，内部設計ができず，設計図なしで頭に浮かんだプログラムをそのままソースに打ち込む方法をとりました。そのため，開発途中に多くの矛盾やバグにぶつかりました。また，スクリプトエンジン（ソフト本体）は一度クライアントにダウンロードされるため，ファイル・サイズを最小にしなければならないという苦労もありました。

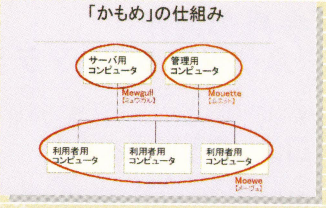
注目してもらいたい点は，プラグインのダウンロードなどの必要が一切なく，現存の技術をそのまま使用する形で使えるというところです。開発者，利用者に一切の負担がなく，面倒な設定も一切なく，Jaguarを使うことができます。さらに，HTMLソース内にテキストでスクリプトが表現されるためJavaScriptやCGIのような他のWeb技術からJagaurを操作することもできます。

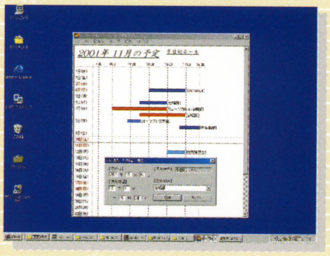
現在，Jaguar1.0は一般には公開されていませんが，言語仕様と解析アルゴリズムを徹底的に合理化し，Jagaur2.0という形で正式公開版として配信しようと考えています。




**作品名** 予約管理システム「かもめ」

神奈川県立神奈川総合高等学校
普通科３年
水谷正慶



**表彰理由：**

学校行事や部活などで，図書館などの施設を利用する際の予約管理ソフト。予約登録・閲覧などをオンライン化することで，端末からの操作を可能にしている。また，上位権限を有する者が強制的に変更可能であるとともに，利用日まで1週間以内であれば変更できないなど，システムの完成度が高い点が評価された。

**作品名** 献立作成表

佐賀県立鹿島実業高等学校
食品調理科３年
釘尾裕美










**表彰理由：**

Excelを用いて作成した献立作成ソフト。性別・年齢などに応じて食べたい食品を入力すると，必要栄養量を計算し，栄養過多の場合は必要な運動量（たとえば散歩10分）を示すなど工夫が見られる。学校で学んでいるテーマを選び，その学習の成果をまとめた高校生らしい作品。

## プログラミング（専門学校生の部）

入賞




作品名 Switch & Shift

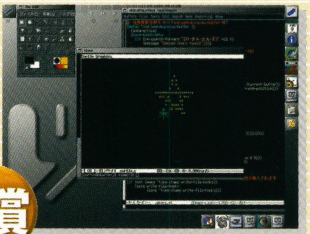
横浜デジタルアーツ専門学校
マルチメディアスペシャリスト科３年
佐々木恵祐　岡田友樹　萩原明人
石原　巧　伊藤直樹　川越紳吾
小池こずえ

**表彰理由：**

携帯電話（iアプリ）に対応したゲームソフト。とくに，20～30歳代をターゲットに電車の待ち時間などのちょっとした時間つぶしを想定している。携帯電話向けに操作性や視認性が良く，うまくまとめられている。

作品名 Curtle Graphics

日本電子専門学校
コンピュータ総合技術科２年
酒井基之



入賞

**表彰理由：**

定められた文法に従ってコマンドを入力することで，画面上のキャラクター（亀）を操作し，図形を描くことができる作品。一見，古典的な内容だが，インタプリタ系の言語（Basic）を学習するツールとしては，実用性が高く，「プログラムとはこういうもの」を体験できる点が評価された。

## コンテンツ（高校生の部）

作品名 もじもじかずかず

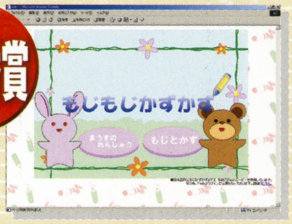
千葉県立一宮商業高等学校
商業科３年
西周由貴　田中優子　露崎俊之　矢澤浩一
情報処理科３年
藍　佑次　石井智子　石川美帆

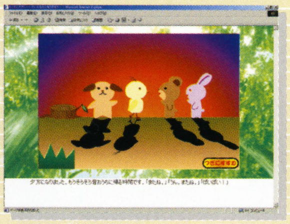


**表彰理由：**

幼年期の子供たちが，ゲーム感覚で楽しみながら読み書きの学習ができるという内容。豊富なイラストと動画を使用して，見た目にも楽しい。正確に画面上の文字や数字をなぞることができると「優しいほめ言葉」が画面に現れるなど，飽きることなく勉強できるよう配慮されている。作成者たちが楽しみながら作ったようすがうかがえる。

最優秀賞



作品名 Seiryo Bunko―図書情報システム―

優秀賞

福島県立清陵情報高等学校
情報電子科３年
片岡優貴　坂倉祥子



**表彰理由：**

学校の図書館すべての蔵書を対象にした検索システム。検索したい図書名や著作者名の一部を入力しても検索できたり，館内の地図や写真など視覚面にも工夫されている。楽しく図書館を利用できるように配慮された実用性の高い作品。




作品名 熊工ホームページ

入賞

埼玉県立熊谷工業高等学校
基礎工学科３年
須田和宏



**表彰理由：**

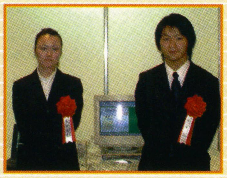
作成者が通う高校の紹介ホームページの出来が悪いと感じ，自らホームページのリニューアルに取り組んだ作品。自分で取材した学校紹介の画像は，デジカメやスキャナで取り込むなど，全体のまとまりもよく，見やすさの点でもうまくまとめられている。

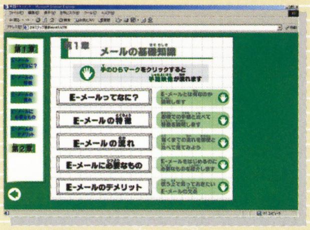
## コンテンツ（専門学校生の部）

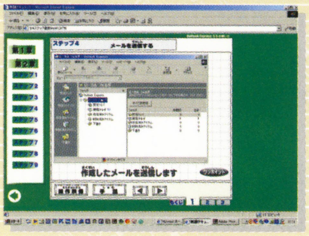
作品名 はじめてのOutlook Express～教育コンテンツのCD-ROM制作～

優秀賞

横浜デジタルアーツ専門学校
マルチメディアスペシャリスト科３年
徳永 聡　藤田美有





**表彰理由：**

聴覚障害者に対するコンテンツの説明が少なく，かつ，文字中心であることを知った作成者が，手話を使ってパソコンの操作方法を教えるために作成した。アプリケーション・ソフトの操作法を文字で解説するとともに，同一画面上で手話（動画）を取り入れ，より聴覚障害者が理解できるような工夫がされている。テーマとしても社会派視点であるなども評価のポイントとなった。

作品名 日めくり家の１年 前編

入賞

静岡産業技術専門学校
マルチメディア科２年
田畑典子　斎藤麻衣子
伊藤大師




**表彰理由：**

国民の祝祭日や記念日，季節の変わり目などを知らせる二十四節気など，日本の伝統をわかりやすく伝えるためにアニメーションで表現した作品。アニメーションや画像を豊富に使い，わかりやすく，親しみやすく，丁寧に作られている。作成者達自らが取材したデータベースもしっかりしている。

# 京都府立 編集部の学校訪問記

# 綾部高等学校

**http://www1.kyoto-be.ne.jp/ayabe-hs/**

〒623-0042 京都府綾部市岡町長田18

東分校　http://www1.kyoto-be.ne.jp/higashi-ayabe-hs/
〒623-0012 京都府綾部市市川糸町堀ノ内18

全日制／普通科第Ⅰ・Ⅱ・Ⅲ（体育系）類，農業科，園芸科，農芸化学科（1099人在籍）
定時制　（58人在籍）　教職員　108人

**綾部高校情報企画部の目的**
校内教職員研修の推進／生徒基本情報の一括管理／校務処理の効率化／ネットワークを用いた学校広報活動／ネットワーク管理／情報教育推進会議の開催／情報教育管理者グループの開催／情報教育アドバイザーとの連携／IT講習会の企画運営

京都府立綾部高等学校は，桑波功一郎校長先生の「学校広報活動を強くしていこう」という方針のもとにホームページ（HP）やメールニュースの配信を強化している高校です。こうした活動の結果，膨大な数の高校生に役立つリンク集が完成し，美術や書道の時間にもパソコンが利用され，政府が実施しているIT講習会では生徒たちも講師補助を行うなど，さまざまな場面でパソコンを活用し，実績を上げています。

HPを立ち上げてから約3年で更新回数は60回，新着記事数は250本以上。目的意識のはっきりとしたHPを自主的に運営しています。今回はこうした活動に積極的に取り組む『頑張る先生たち』をご紹介します。



▲綾部高校本校校舎

## 登場する先生



桑波功一郎校長



今井裕子先生（書道担当）



大音　愛先生（美術担当）



高橋光重先生（綾高教育リンク集責任者／理科担当）



大西勝己先生（情報企画部部長／数学担当）



田中文浩先生（校内LAN管理者グループ責任者／数学担当）

## ホームページ／メールニュース配信など「頑張る先生たち」

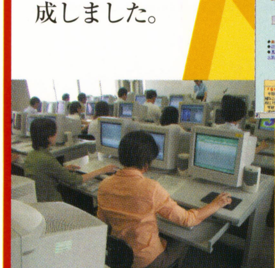
### ホームページの目的と歴史

「綾部高校のHPは『単なるHPづくり』を目的としてはいません。インターネットを活用して学校広報活動を行うことを明確な目的としたHPです。小中学生とその保護者に正しい情報の中で高校を選択してもらうための，信頼あるHPづくりが必要である，と考えています」と話してくれるのは大西勝己先生です。大西先生は学校広報を目的としたHPづくりに取り組むための校内組織，情報企画部の部長も務めておられます。

1998年8月，綾部高校に40台のWindows95搭載パソコンが導入されました。これをきっかけに先生たちは校内で「情報教育推進会議」を結成。30人の先生たちが参加しています。さらに「HP編集グループ」「情報教育（授業）推進グループ」「校内LAN管理者グループ」「校務処理推進グループ」が下部組織として結成されています。内部でのHPの検証などを行い，12月1日に正式に「綾部高校HP」が完成しました。



▲綾高の広報活動に大きく貢献しているHP



▲教職員を対象にしたExcel研修のようす。こうした教師の研修のようすはHPで公開





▲学校案内と卒業記念のCD-ROM

### 編集会議で更新内容を吟味

「HP編集会議は月に一度開いています。この会議には教頭先生や教務部長も参加します。会議で学校全体としてHPへの掲載内容を決定するためのものです」と大西先生。原案は大西先生が提出しますが，多くの先生がHP編集に携わるようにしています。

会議で話し合う内容は「どこにニュースバリューがあるか」ということです。HPに掲載することで校内にやる気が出る，そんな内容をみんなで探そうとしています。

HPに掲載する内容は何がよいか？という判断には経験を積んだ先生が必要です。そこで登場するのが田中文浩先生です。田中先生は有志を募って校内LANを張り巡らし，校内ネットワークをつくりあげた実績の持ち主です。

デジカメで部活動や校内行事などを撮影し，さまざまなHP向けの素材を保管しています。こうした中からHPに掲載される内容が決定されていきます。

HPは「毎月更新」が前提です。更新内容はさまざまです。クラブ活動の紹介から卒業式のようす，進路実績，PTA，お便りコーナーなど幅広い内容の更新が行われてきました。「スキー研修現地速報」などリアルタイムで更新される内容もあり，先生たちのHPへの取り組みぶりと共に，期待を寄せる保護者たちの高い関心もうかがえます。

### CD-ROM制作にも力を注ぐ

よりたくさんの情報を伝える目的で学校案内をCD-ROM化しています。京都府の府立高校の中では初めての試みだそうです。卒業記念アルバムのCD-ROM版を製作し，卒業生に渡している点も特徴的です。

綾部市の中学校ではパソコンの導入が一斉に行われました。中学生に向けたPRの手段として，ますます，HPやCD-ROMの活躍の場が増えていきそうです。

## メールニュースと中学生向け掲示用ニュースの発行



▲デジタル情報以外にも情報誌を“紙”で発行

綾部高校のもう一つの特徴はメールニュースの配信です。HP上に登録ボタンが設定されており，ボタンを押すことでメールニュースが配信されるようになります。卒業生を中心に現在400件以上の登録があり，情報を必要とする人にHP更新情報や学校行事などを定期的に配信しています。

また，中学校に掲示してもらう目的で「さわやか通信」と「翔綾」を紙で発行しています。

## IT講習会で活躍する生徒たち

綾部高校は京都府の「IT講習会」の講習会場にもなっています。綾部高校は「教員の手づくりの講習会」「生徒の力を借りる，もしくは発掘する講習会」にすることを決めて講習を開始しました。講師と補助の先生が合わせて4人，補助の生徒が3～4人，という体制で第2回から継続しています。

生徒の力を借りることになったきっかけは，ある卒業生からのメールでした。文章に添えて綾部高校の校歌がMIDIファイルで添付されていたのです。素晴らしい出来映えに「僕たちの知らないパソコンの達人が生徒の中にいるのではないか？という興味が湧きました」と大西先生。講師の募集を呼びかけたところ，「協力してもいい」との返事をしてくれた生徒たちが補助として手伝ってくれています。生徒たちが手伝うことで，講習の雰囲気が全くちがったものになり，場が和らぎます。受講者からは感謝のメールが届き，生徒も教える立場について深く考えるようになります。講習は京都府のテキストを利用して進められますが，綾部高校独自の「おみやげ」も用意しています。普段の講習風景を撮影しておき，最終日に撮った集合写真をサーバにアップします。この写真を受講者は自分でフロッピーディスクに取り込んで完成させる，という実務のともなう「おみやげ」です。HTMLでHP風に仕上げられるように工夫されており，BGM付きという立派な内容。マルチメディアの一端を見てもらうことが目的です。

「生涯学習とはこういうことか，と実感しました。IT講習会によって学校という壁が取り払われ，地域が学校を，学校が地域をお互いに受け入れ合う，開かれた学校，という状態が実現しました」（大西先生）。

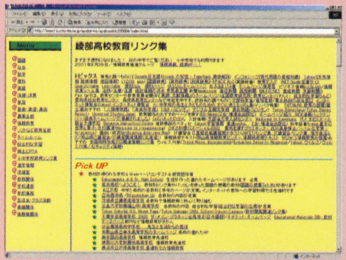


▲国で行われているIT講習会の会場にもなっている綾部高校。生徒も講師補助として活躍

## パソコンを生かした授業の数々

### Link集を作成／高橋先生

小中学校用のリンク集は数多くあるのですが，高校生向けのリンク集が少ないとのことで，2年ほど前から検索エンジンを使って高校生に役立つHPを見つけて登録していく作業をはじめました。今では2,000から3,000件のHPは登録されているのではないでしょうか。他の高校の先生や生徒にも使ってもらいたいと思っています。『情報』が必修科目になったときに役立つはずです。




▲高校生に役立つHPリンク集は圧巻！「他校にもどんどん使ってもらいたい」

### 書道／今井先生

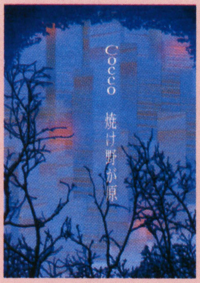
写真と文字を合体させた作品づくりに取り組んでいます。自分が撮影した写真もしくは自分がモデルになった写真と，自分で書いた「漢字かなまじり」の書をパソコンで合成して作品に仕上げます。生徒自身の意図を超えた芸術が生まれることもあります。一見，醒めているように見える生徒たちも，自分のメッセージや夢を表現することで，心の奥に持っている熱い思いを垣間見せてくれます。昨年は写真はよいのですが字が未熟という傾向が見られたので，今年は字の勉強に力を入れました。ただし生徒が喜ぶから，という理由だけでパソコンを使っていては本来の書道教育から逸脱しますので，あくまで応用的なものとして書道Ⅲの選択者を対象に取り組ませています。



▶書道と写真を融合させた生徒作品

### 美術／大音先生

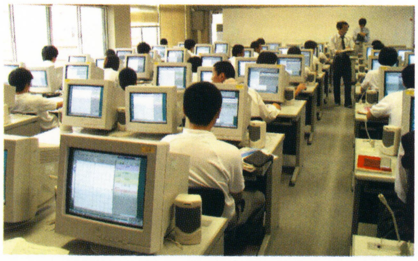
美術Ⅲ選択者を対象に音楽のCDジャケットをCGでつくっています。曲を伝えることを目的に雑誌や写真集からの切り抜きや合成などを利用して，何度もやり直しを効かせながら，手を加えて作品を完成させていきます。生徒たちの「これをやりたかった」という部分を引き出せたと思います。しかし，筆描きに比べ簡単に作品ができますので，あとから筆で描くのがむずかしくなってしまう，というデメリットも意識しなければなりません。




▶CDジャケットをCGで作成した生徒作品

## 必修教科「情報」開始に備えて

平成15年4月から高校の授業では「情報」という教科がスタートします。一方でそれを待たずとも，生徒たちはすでにパソコンを使いはじめています。「生徒へのアンケートの結果，自宅にパソコンがある家は全生徒の3分の2，うち半数がインターネットに接続できる環境にありました。情報不足なのはむしろ親や教員ではないでしょうか」と田中先生。正しい情報との付き合いかたを知る，それは単にインターネットやパソコンの技術的な知識を知ることではありません。



▲生徒のための「情報教育ガイダンス」風景。ネチケットに対する取り組みも真剣に行われている

綾部高校ではインターネット上での礼儀や正しい使いかたといった意味を持つ「ネチケット」の問題に取り組んでいます。生徒たちに対しては新入生を対象に毎年5月，コンピュータガイダンスを実施します。今年は出会い系サイトについて注意を呼びかけました。教職員研修としては「インターネットの光と影」と題した講座を設け，とくに影の部分について知識を深めました。

「ネチケットの問題は情報教育のうえで一番の問題だと言えます。よい部分を見ると，インターネットは世界中をデータベースとしてしまうことが可能ですね。1個人が生涯の間に知り得る情報の量をインターネットは拡大していきます。こんなすばらしい道具を利用しない手はない。私の担当する数学でパソコンは思考力を養う手助けをしてくれるでしょう。しかしプライバシーの侵害など影の部分を知ることも大事なことです。文部科学省の方針では，すべての教室はインターネットに接続できる環境を整えていくことになっています。綾部高校では本校と分校あわせて9人の先生が『情報』の教員免許を取得し，来たるべき時期に備えています」と大西先生は話してくれました。

## 今後もますます期待が高まる

HPをつくったり，CD-ROMを作成したり，これまでに紹介してきた綾部高校の活動は，これからより複雑化していく情報社会に正しく対応できる生徒たちを育てていくための取り組みだと言えますね。みなさんの身近にもきっとこうした問題に取り組もうとしている『頑張る先生たち』がいるはずです。



▼2年1組の生徒たち。「このクラスのレベルは高い」と先生の評価

# 長野県 編集部の学校訪問記
# 豊科町立豊科北中学校

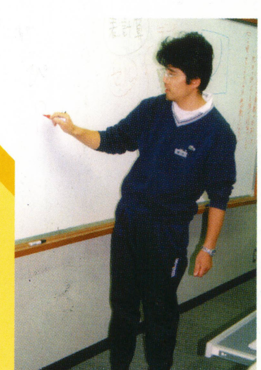
http://www.town.toyoshina.nagano.jp/jhs_kita/index.html

## Excelを使用した授業を拝見！ホームページでの情報交換も活発！

### クラブ活動が活発な学校

信濃の国・長野県のほぼ中央に位置する豊科町。豊科町立豊科北中学校は，周辺が田園に囲まれ，北アルプスがのぞめる豊かな自然環境のもとにあります。豊科北中学は開校してからまだ17年目ですが，新制中学の創立時までさかのぼればじつに55年目を迎えました。

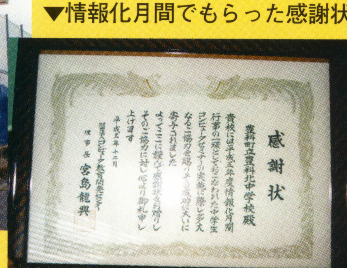
クラブ活動が盛んで，吹奏楽や陸上，ソフトボールなどで輝かしい成績を残しています。なかでも，女子ソフトテニス部は長野県や北信越地区ではトップレベルにあり，今年も全国中学校ソフトテニス大会に出場（個人戦）するほどの実力です。ちなみに，豊科北中女子ソフトテニス部はホームページを持っており，理科の丸山毅先生曰く「学校のホームページより後にできたのに，アクセス数で抜かれてしまいました」と苦笑いしていました。



▶高校でも教鞭をとっていた野本先生



▲豊かな自然環境の中に立地する校舎。手前が全国レベルの力を持つテニス部のテニスコート




▼情報化月間でもらった感謝状

### 着実に情報教育に取り組む姿勢

パソコンなど情報教育にはこれまでも力を注いでおり，平成5年に生徒用のパソコン20台，教師用2台の計22台を導入し，コンピュータ教室を稼働させました。同年12月には平成5年情報化月間行事の一環として行われた「中学生コンピュータセミナー」の実施に際し，財団法人コンピュータ教育開発センターから感謝状を受けました。

そして，平成10年10月に現在のパソコン教室が稼働開始し，マルチメディア・インターネット対応となりました。担当教諭の丸山先生は「新パソコン教室にはパソコン（エプソンTYPE-FR）が40台入っています（他に先生用1台）」と話し，生徒1人に確実に1台が使えるようになりました。

同教室には周辺機器として，A4カラー・プリンタ5台（ネットワーク），A3カラー・プリンタ1台（同），A4レーザー・プリンタ2台（同），B4レーザー・プリンタ1台（同），スキャナ，MO，CD-R，VTR，OHC（掲示取り込み用），液晶プロジェクタなどが完備しています。

### 初めてのパソコン授業を拝見

取材時に，2年生の技術・家庭科の授業を見せてもらいましたが，パソコンを使った授業はこれが初めてとは思えないほど，生徒たちはマウスを器用に動かしていました。技術・家庭科の野本裕之先生は「クラスによって授業の進みかたが早かったり，遅かったりしますが，ペースにあわせて進んでいきます」と余裕のあるコンダクターぶりです。それもそのはず，先生は昨年まで池田工業高校に勤務していましたが，今年から中学校と高校の交流の一環として，中学校で教えることになったとのことです。

この日の授業で勉強していたのは，Excel（表計算）。Excelと言えば，初歩をある程度マスターしてからパソコン教室などで学ぶレベルですが，最初の授業でいきなりExcelとは，昨今の子供たちのレベルは高いようです。教材には履歴

# 新パソコン教室での技術・家庭科の授業風景



▲今回は初めてパソコンを使っての授業だけに多少緊張気味



▲先生用のパソコンから見た風景。みんな一生懸命だ



▶わからないときは協力し合うことでコミュニケーション能力の養成にも



▶熱心に履歴書を作成する生徒たち

書を使用し，各人が履歴書を作ってみようというものです。野本先生が「履歴書に使う写真は，デジタルカメラで自分の顔写真を撮影し，取り込んでみましょう」と話すと，生徒たちはみんな，すごい興味を示していました。

パソコンの操作に抵抗のない生徒が目立ちますが，丸山先生によれば「パソコンを家庭で使っている子供は3分の1程度。ただし，ゲームなどで楽しむことが多いようです。HPを自分で作るぐらいの子供は，私が知っている限りでは1人だけです」とのことでした。

取材当日の技術・家庭科の授業でも，入力のままならない子供や，まわりの友達の助けを求める姿も多く見られ，なぜか少し安心しました。技術・家庭科のパソコンの授業は，2年生が半年間（週2時間），3年生も同じく半年間（週1時間）です。従来は，3年間で半年間だけだったので，パソコン教育の可能性が広がりました。「なんとかHPの作成まで学習させたいのですが」とは野本先生の弁です。

## ビデオ・オン・デマンドも活用

パソコンの利用は，技術・家庭科以外にも「音楽（ミュージックドリル）や美術（スキャナを活用），理科でも使っています」（丸山先生）。理科の授業では，VOD（ビデオ・オン・デマンド＝見たいときにいつでもテレビ番組が見れること）を今年の8月から活用すべく取り組み，校内VODとしてNHKテレビの理科の番組の視聴に活用しています。

生徒によって，テレビ番組を繰り返し見たいという要望も多く，静止させたり，前に戻って見るなど，VODのメリットを生かしています。また，VODは豊科町が取り組みを強化しており，欅の家（中間教室：不登校傾向にあり，学校へ行けない生徒が通う施設）や小学校への配信にも取り組んでいます。




▲女子ソフトテニス部のホームページ。戦績や部員の活動ぶりがよくわかる

今後のパソコンの活用について「せっかく校内のネットワーク（LAN）が構築されているので，いろいろな人に見てもらいたい。そして，お互いに理解しあえたらいいですね」と丸山先生と野本先生の意見は一致しています。

## ますます充実するホームページ

豊科北中学校のHPは，平成11年7月に開設。アクセス数は平成12年9月からのカウントで1万を突破しています。ユニークなのは，「北中ネットワーク」。全国の〇〇北中学校がネットワークを持ち，情報交換しようというもの。卒業生の部屋や豊科通信，そしてメールマガジンもあります。iモードでも行事を見ることができ，便利です。




▲学校生活や年間行事予定，生徒の作品などもおさめられたホームページ

**豊科北中学校** 〒399-8205 長野県南安曇郡豊科町大字豊科5558 TEL0263-72-2265

発売が間近に迫った次世代 Windows システムの注目機能と
インストール& PC アップグレード・テクニックを徹底紹介!!

# Windows XP 導入実践講座

パソコン界で今年のトピックといえば，Windows XP の登場があげられるでしょう。Windows MeやWindows2000の後をつぐ次世代OS，Windows XPは，マイクロソフト社のOSを統合し，マルチメディア機能の強化やブロードバンドに完全対応した最強のOSといっても過言ではありません。このページでは，XP の新機能やインストール方法，さらにパソコンのパワーアップテクニックまでWindows XP導入に役立つ実践活用法をまとめてみました。

## 1 どこが変わった…？　XP の操作方法と追加機能

マイクロソフト社のWindows OSは，Winows 98／Meのコンシューマ（一般ユーザー）向けとWindows 2000のワークステーションすなわちビジネスユーザー向けの2系統のOSが用意されていました。今回のWindows XPでは，この2種類のWindowsを統合し，32ビット完全対応OSとして登場したわけです。

さらに，従来とは大幅に変更されたユーザーインターフェース（UI）も大きな特徴となっています。Windows XP の「XP」は，“eXPerience”（体験）の略で，

標準Webブラウザは，インターネットエクスプローラ6.0にバージョンアップ。プライバシー保護機能の強化や動画再生対応のエクスプローラバーが追加

左ペイン（窓）のデザインが変更され，新機能が追加

新しいユーザーインターフェース「Luna」の採用でパステルカラーを多用し，見栄えのする楽しげなWindows画面に変身!!

2列表示に変更されたスタートメニュー。表示項目が増え，アプリケーション起動や各種設定などが，素早く行えるようになった




画像ファイルを保存したマイフォルダなどでは，従来の縮小版表示のほか拡大版表示機能が追加された。画像の90度回転などもここから直接行える

表示内容に合わせたタスクが自動表示されるウインドウの右ペイン（窓枠）

従来よりも大きくなったスタートメニューのボタン

アプリケーションごとのグループ化が可能になったタスクバーでウインドウのスムーズな表示切り替えを実現

常駐アプリのインジケータの表示／非表示は境目のボタンのクリックで選べる

新しいWindows OSで強化された魅力的な機能を広くパソコンユーザーに体験してもらいたいことから付けられたそうです。マイクロソフト社が満を持してWindows XPを開発した意気込みがうかがえます。

まず，Windows XPで目を引くのは，起動後に現れるデスクトップ画面でしょう。スタートメニューや画面下にあるタスクトレイは，以前のWindowsでおなじみのスタイルですが，操作方法や機能はこれまでとは比較にならないほど，強化&変更が施されています。

Windows XPでは，OS操作の基本となるユーザーインターフェースに，新しく「Luna」インターフェースを採用しています。

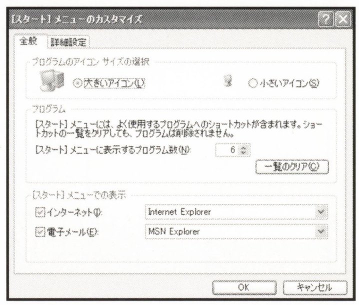
コーナーに丸みを帯びたウインドウやパステルカラーを使った配色，カーソルのポイントで色が変わるタブやボタンなど，これまでのOSとは，ひと味違うインターフェースデザインになりました。実用一点張りから操作が楽しめるOSに進化した点は好感が持てます。

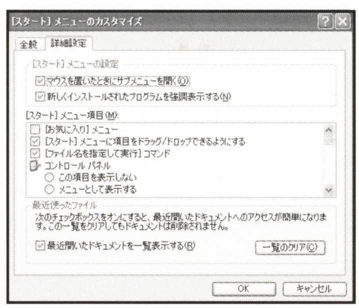
驚いたことに，Winows XPを新規インストールすると，デスクトップ上に，ごみ箱以外のマイドキュメントやマイコ

## すべての操作&設定が行える多彩な機能を持つスタートメニュー



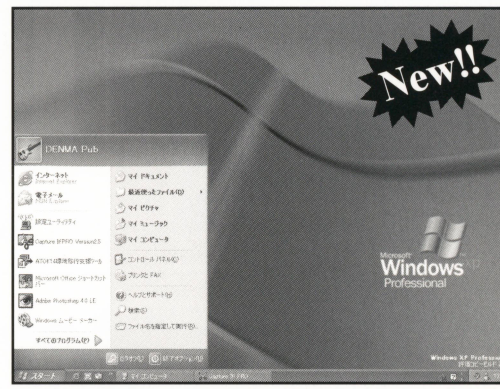

▲スタートメニューは，２列配列のメニュー配置に変更。左側の列は，固定（上側）と使用頻度で入れ替わる自動表示のショートカット・アイコンが並んでいる。固定アイコンは，希望のアプリケーションへの変更が可能

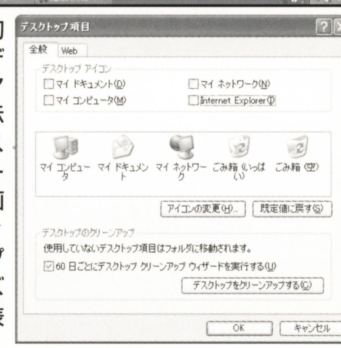



▲表示メニュー，アイコンはコントロールパネルのスタートメニューのカスタマイズで変更できる

## デスクトップからフォルダ&ファイルを排除

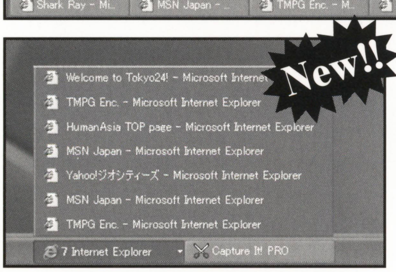


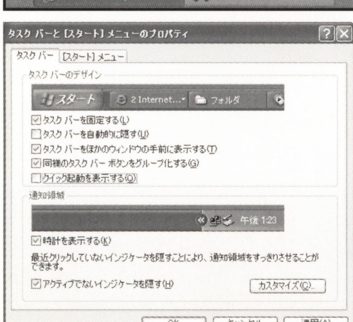

デフォルト（初期状態）では，デスクトップのアイコンは非表示だが，操作はスタートメニューから行える。画面のプロパティ→デスクトップのカスタマイズ（▶）の設定で表示可能

## グループ化対応のタスクトレイ




◀タスクバー・エリアにボタンが収まりきれなくなると自動的に同じアプリケーションのボタンにグループ化される。グループ化のオフも可能だ




◀タスクバーの設定は，スタートメニューと同じプロパティ画面のタスクバー・タブから行う




▲常駐ソフトの表示インジケータは，境目の丸ボタンのクリックで非表示にできる

## 縮小版表示と拡大版表示が選べる画像ファイル




◀マイピクチャなどのフォルダ内に画像ファイルがある場合は，プレビュー画像が縮小版表示のほか拡大表示も可能になったほか，フォルダアイコンは保存ファイルのプレビューアイコンで表示

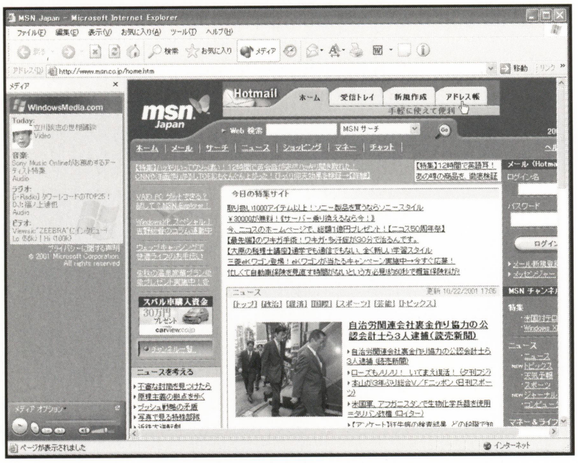


▶従来どおりに縮小画像のサムネイル表示も可能

ンピュータなどのアイコンが見あたりません。フォルダの操作機能がスタートメニューに組み込まれ，デスクトップにたくさんのアイコンがあるために起こる操作のしにくさを軽減できるメリットがあります。もちろん，従来どおりのアイコン表示も設定可能です。

Windows XP のスタートメニューは，2列表示になって表示項目が増えています。プログラム起動をはじめ各種設定項目が選択できるスタートメニューは，OS操作の中心部ともいえるほど，重要な役

## 標準 Web ブラウザは IE6.0 へバージョンアップ

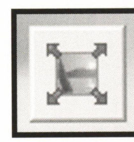




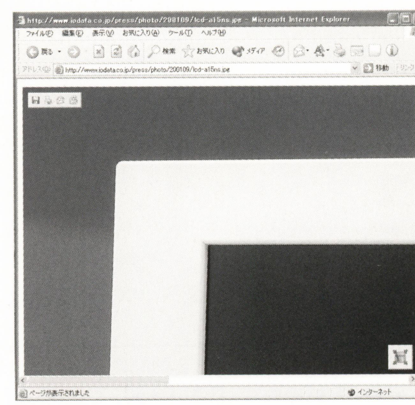
画像表示部分に表示されるツールバー。イメージ保存，印刷，メール添付，マイピクチャ・フォルダを開く，が可能になる



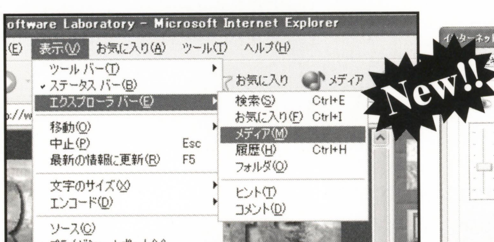

◀機能強化でさらに使いやすくなった。左のエクスプローラバーで音声や動画再生が，ブラウザだけで行える



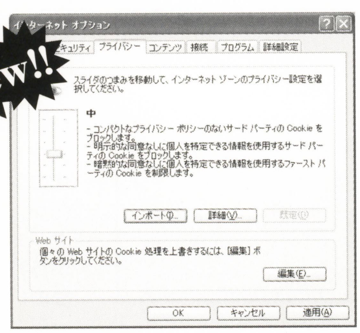
イメージ本来のサイズでの表示に切り替えるリサイズボタン



▲Webブラウザの画面に収まる縮小サイズに自動調整するリサイズ機能がある。ボタンのクリックで拡大表示が可能



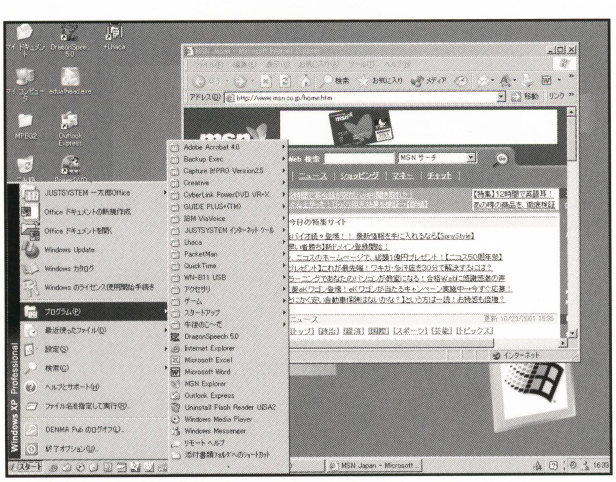

▲マルチメディア機能の強化も特徴のひとつ。表示メニューの「メディア」で，音楽&動画再生が，エクスプローラバーから可能



◀ Cookie によるハッキングを防ぐプライバシー保護機能が「P3P」対応となり，サーバごとの細かなプライバシー保護設定が行えようになった

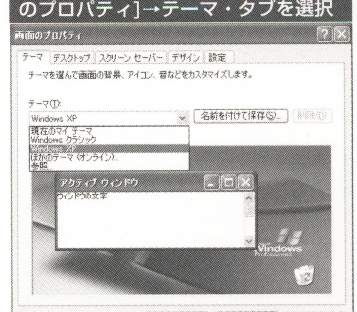
## XP を Windows Me ／ 2000 のデスクトップ・スタイルで使う!!

Windows MeやWindows 2000からの乗り換えで気になるのは，Windows XP でがらりと変わったユーザーインターフェースになじめるかでしょう。アイコンのないデスクトップなど，これまでのOSとは少々違った操作が求められるのも事実。Windows XPは，「Luna」インターフェースのほかに「クラシック」インターフェースが用意されています。このクラシックを選ぶと，Me や 2000とほとんど同じインターフェースを持つ Windows XP に変身させることができるので安心です。

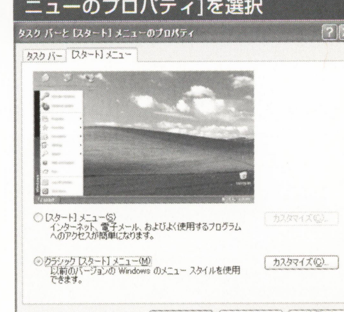


▲「Luna」デザインをWindows MeやWindows2000のクラシック・スタイルへ切り替えることも可能

デスクトップ上でマウスの右ボタンをクリック→プロパティを選択→[画面のプロパティ]→テーマ・タブを選択



[スタート・ボタン]→[コントロールパネル]→[タスクバーとスタートメニューのプロパティ]を選択

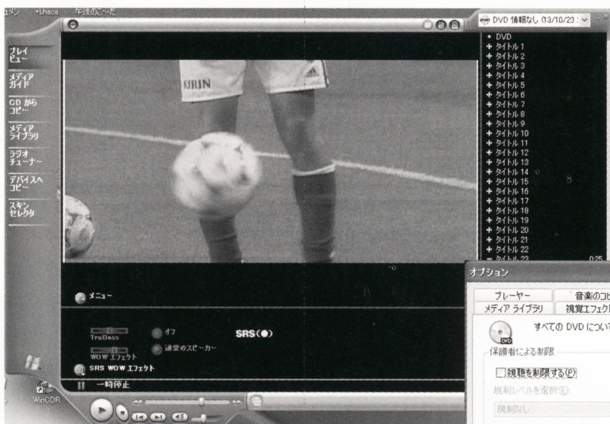


ユーザーインターフェースを自由にカスタマイズできるのもWindows XPの特徴。左のクラシック環境に切り替えるには，「画面のプロパティ」(左の設定画面)と「タスクバーとスタートメニューのプロパティ」(右の設定画面)で，クラッシックを選べばOK

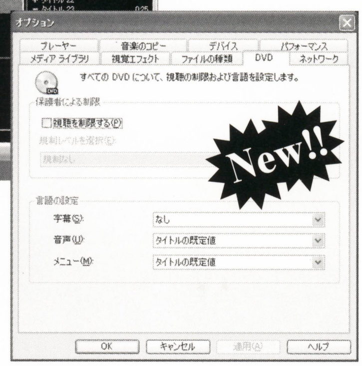
割を持っていますが，設定項目を増やすことで，これまで以上のスピーディな操作が行えるようになりました。

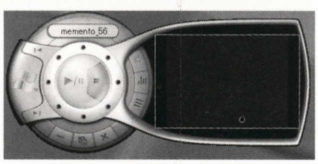
そのほか，DVD 再生にも対応したメディアプレーヤーや動画のコンテンツ表示や再生が行えるメディア機能が強化されたインターネットエクスプローラ 6.0 の装備など，Windows XP は最強の OS の名前に恥じないさまざまな数多くの機能強化がはかられ，以前の Windows とは別物といってよいほどの充実した内容になっています。

## 機能強化がはかられたメディアプレーヤー&ムービーメーカー

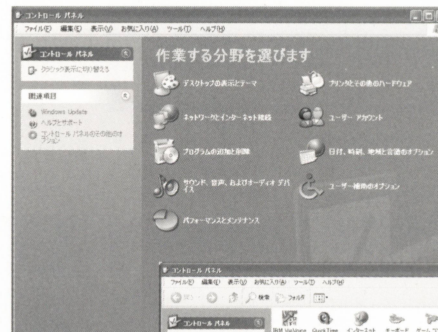


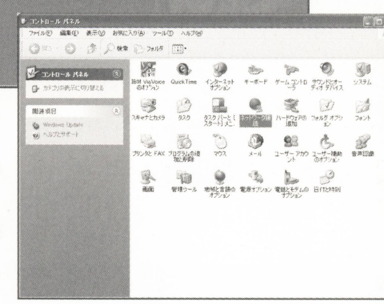
◀Windows Meから標準装備となったメディアプレーヤーは，XP バージョンにアップ。DVDビデオ再生や音楽CDライティング機能を追加



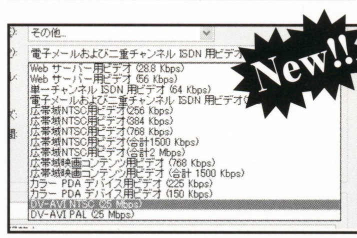



▶ボタン配列やデザインが選べるスキンの種類も増えている







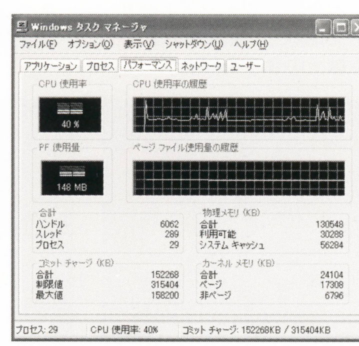



◀▲ビデオカメラとの連携で手軽にビデオ編集が行えるムービーメーカーは，DV-AVI形式でのファイル保存機能が可能になった

## カテゴリー化されたコントロールパネル

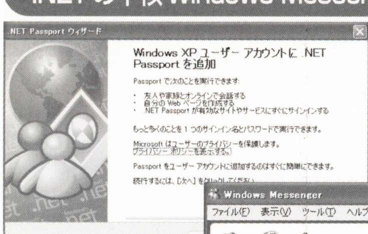
ハードウエア設定を行うコントロールパネルは，デバイスのカテゴリーごとに整理され使いやすくなっている。ユーザーアカウント（登録）もここから行う

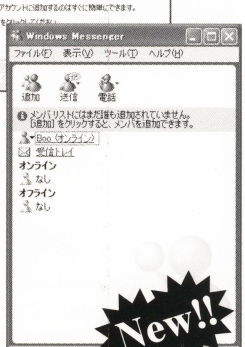
▶左ペインの「クラシック表示に切り替える」ことで，従来同様の操作も可能




◀Windows タスクマネージャからCPUの使用率やメモリ管理状況などの動作状況がモニターできる

## .NET の中核 Windows Messenger

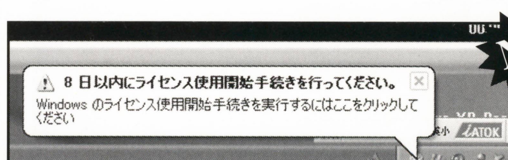




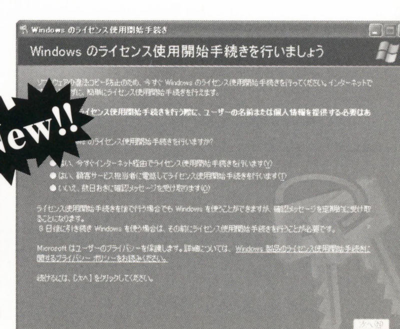

ネットワーク環境での機能強化もXPのウリ。「.NET」サービスを利用し手軽にチャットが行えるWindows メッセンジャーを搭載。音声のほかビデオ映像でのチャット（電話）にも対応。

## 導入後 30 日間以内にライセンス登録を!!

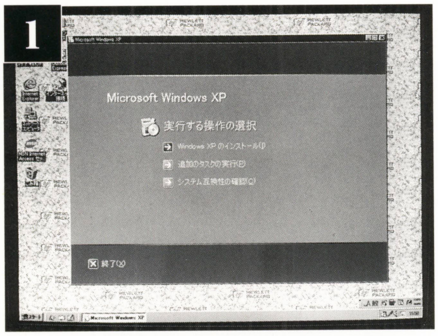
Windows XP では，違法コピー防止のための「アクティベーション」とよばれる「ライセンス使用開始手続き」機能が採用されています。XP導入後にマイクロソフト社へこの手続きを行うことで，導入マシン以外のパソコンへの XP 導入は行えなくなる仕組み。必ず台数分のパッケージ購入が必要です。インターネットからも自動で行えるので，インストール後は速やかに手続きを済ましておきましょう。




▲タスクトレイのインジケータ・エリアにある鍵マークのクリックで「ライセンス使用開始手続き」が行える。導入後30日間のうちに済まそう



# Windows XPをインストールしよう

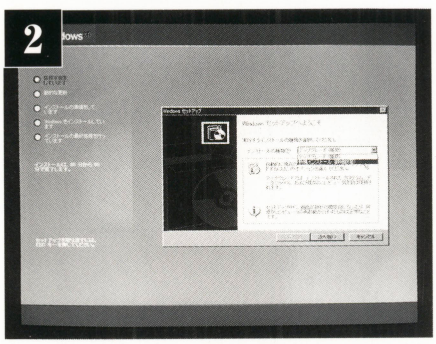



▲旧OSからWindows XPへのアップグレードは，インストールCDをCDドライブへセットすると自動的に始まる

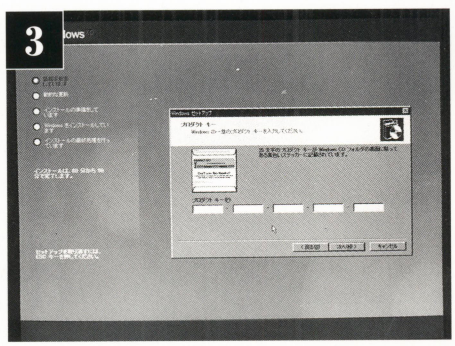
Windows XPは，「Professional」版と「Home Edition」版があり，双方の差は接続先パソコンの操作が行えるリモートデスクトップ・クライアントなどのネットワーク関連やセキュリティ機能の強化のちがいです。

そのほかのマルチメディア・ツールやOS基本機能は共通で，一般的な環境では，安価な「Home Edition」版で十分です。Windows2000やWindows Meからの乗り換えは，アップグレード版を選べば良いでしょう。

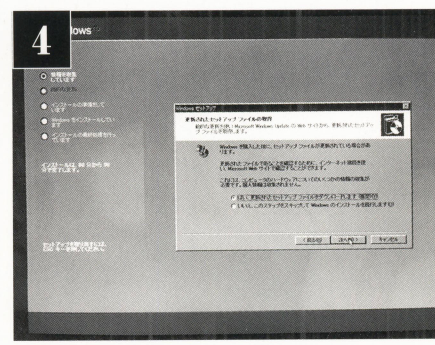
インストールの手順は，どちらもほとんど同じで，画面の指示どおりに進めていけば，問題ないはずです。注意点として，インストールの終盤で行うアドミニストレーター（システム管理者）のパスワード設定があります。インストール後にOS設定などの変更を行う際に，アドミニストレーター資格でログインしなければなりません。パスワードは，必ず覚えておくかメモを残しておきましょう。



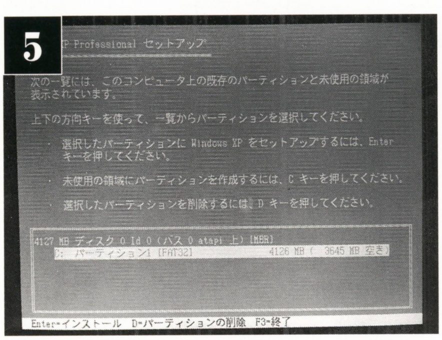
▲アックグレード，または新規インストールからアップグレードを選択。以前のOS設定やアプリケーションはそのまま残せる



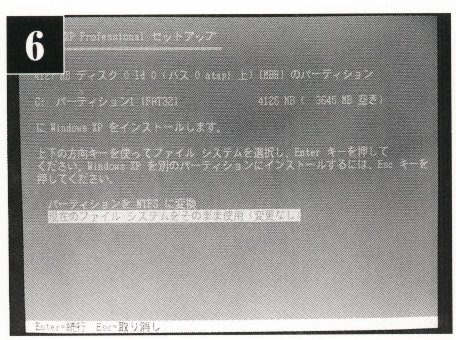
▲インストールCDのケースに用意されたプロダクト・キーを入力する



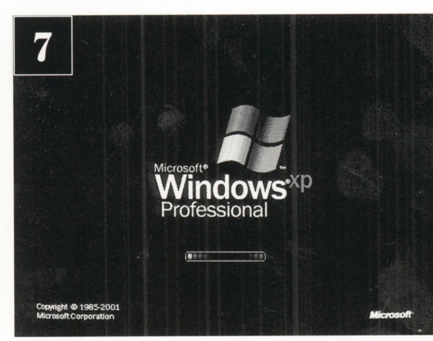
▲インターネット接続が可能な場合，「更新されたセットアップ・ファイルの習得」画面でセット・アップファイルをダウンロードを選ぶ



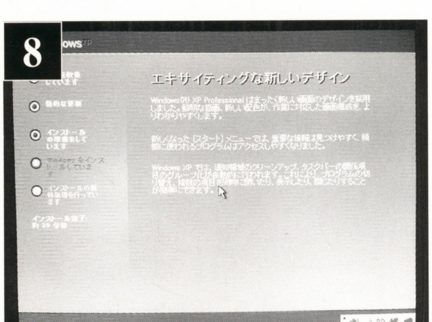

▲ハードディスクにパーティション設定を行っている場合は，XPをインストールするドライブを選択



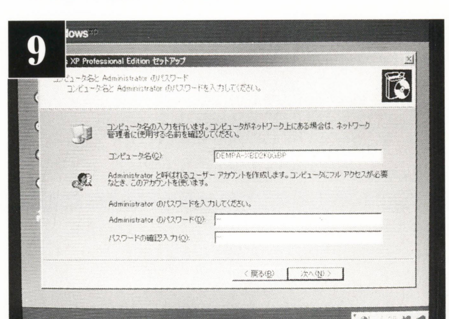

▲ハードディスクのファイル管理システムを「NTFS」に変更するか，これまでの「FAT32」のままを使うかを選択




▲Windows XPインストールの再起動画面。導入後の起動時にも同じ画面が現れる




▲Windows XPの導入に必要なファイルのインストールを開始。機能紹介のスライド画面が表示されるので，要チェック!!




▲アドミニストレーターのパスワードを入力。OS設定の変更などには，アドミニストレーターでのログインが必要




▲インストールが終了し，ログイン画面で先に登録したアドミニストレーターのパスワードを入力し，Windows XPを起動

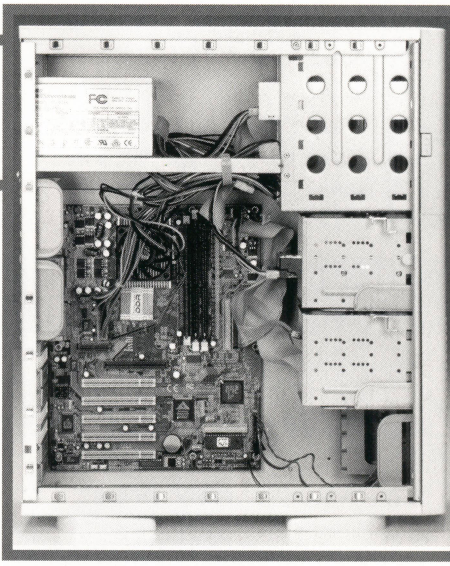
# 3 Windows XP 導入とPCのアップグレード

多彩な機能が追加されたWindows XPは，これまでのWindowsにもまして，パソコンに高い処理能力が不可欠です。右の表は，Windows XPを導入するための推奨スペックですが，Windows Meでは，CPUがPentium 150MHz，64MBメモリ以上のDOS／V機とあることから，XPへの移行には倍以上の性能が必要なわけです。手持ちのパソコンのCPUが，推奨スペック以下であれば，XP対応機を自作または購入するするしか手がありませんが，それ以上のCPUであれば，メモリの追加などのアップグレードで済みます。

ここではWindows XP導入のためのパソコン強化テクニックをいくつかの例を挙げてみました。パソコンの使用環境や予算に合った方法を見つけてください。

●Windows XPシステム・スペック

| | |
|---|---|
| CPU | 300MHz以上のCPUを推奨<br>PC/AT互換機のみ<br>※Professionalは2個，Home Editionは1個までのCPUを搭載したシステムに対応 |
| メモリ | 128MB以上のRAMを推奨<br>※96MB以下の場合，一部の機能に制限がある |
| ハードディスク | 1.5GB以上の空き容量 |



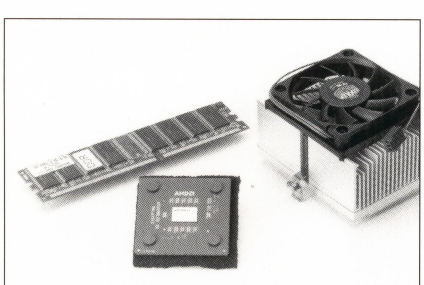
## Windows XPの機能を生かす DOS/Vパソコンの自作にチャレンジ!!

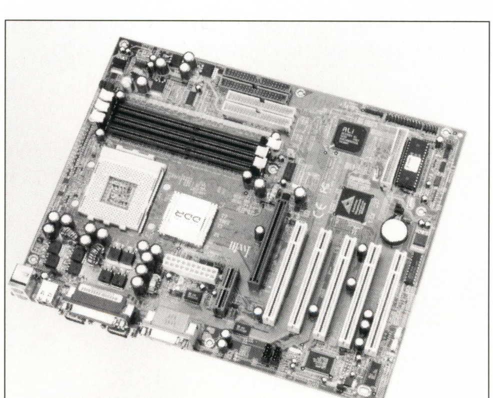
メーカー製のWindows XP対応機を選んでもかまわないのですが，予算が足りないという場合は，DOS／V機の自作にチャレンジしてみてはいかがでしょう。自作ならメーカー製と同じ性能のDOS／V機が，格安で手に入れることができます。ワークステーション・クラスのハイエンド機から安価なコンパクトPCまで，用途や予算に合わせ，世界に1台しかないオリジナル・マシンに仕上げることができるのも，DOS／V機自作の醍醐味といえるでしょう。

Windows XP向けにDOS／Vを自作する場合，まず大まかなスペックを決め，それに見合うパーツを選択します。コストパフォーマンスを考慮すると，1GHz前後のCPU，メモリは256MB，ハードディスクは最低でも10GBが目安です。もちろん予算に余裕があれば，より高速で大容量のパーツを選んでください。とくに，CPUには思い切って予算を十分にとっておいてください。各パーツの選択基準は，つぎのページで紹介していますので，そちらをぜひ参考にしてください。このクラスであれば，安価なパーツを探せば，キーボードやマウスを含めて6万円前後で手に入るはずです。

### 自作DOS/V機，最強の組み合わせは!?

インテル社のPentium 4，AMD社のAthlon XPと，Windows XPの発売に合わせ，高性能CPUが登場していますが，なかでも一番のオススメは，Athlon XP。クロック速度ではPentium 4が一歩先を行っていますが，処理性能では同等かそれ以上のポテンシャルを持ち，価格が安いのも特徴。従来のメモリの2倍の速度で読みが可能なDDR SDRAMとの組み合わせで，最強のDOS／V機を手に入れることができます。





▲▶Windows XPを最強のマシンで，という場合は，最新CPUのAthron XPとDDR SDRAMメモリの組み合わせがベスト。対応マザーボードも続々登場している

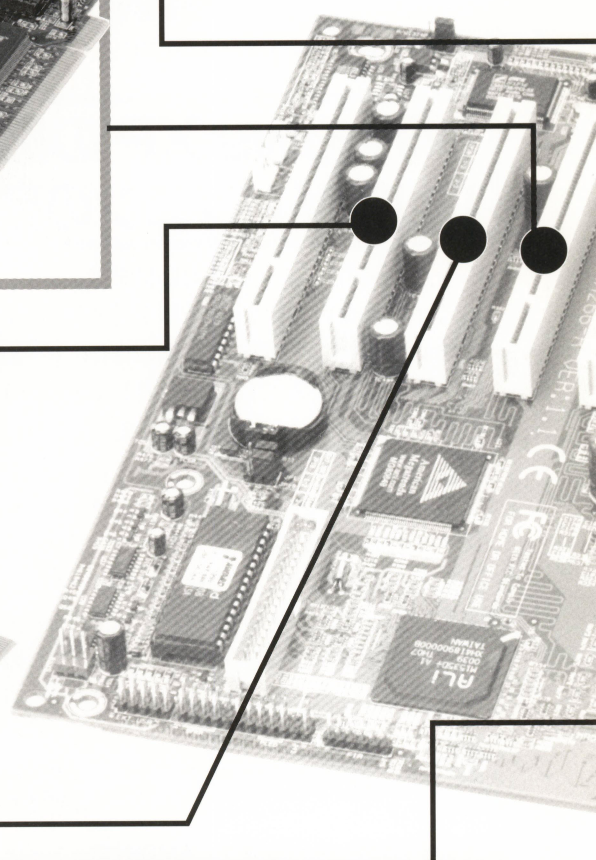
DOS／Vの自作では，CPUとマザーボードは，必ず一緒に購入しましょう。CPUには，AthronやDuronのAMD系とCeleronやPentium Ⅲ／4のインテル系にわかれます。双方のCPUにはハードウエア上の互換性がなく，専用のマザーボードを用意しなくてはなりません。また，マザーボードごとに，メモリの種類や利用できるインターフェースに差があります。CPUとマザーボードを決めることで，パーツも決まってしまうわけです。またCPUとマザーボードやPCIカードとの相性が悪く不安定な動作を招くこともあります。このようなトラブルを避けるためにも，ショップなどであらかじめ相談にのってもらい，実績のあるCPUとマザーボードの組み合わせを選んでください。

## Windows XP 導入のための
# DOS/V PC 自作パーツの選びかた

Windows XPでは，マルチメディア機能やネットワーク機能の大幅な強化が行われています。そのため，ビデオカメラやブロードバンド回線，さらにハードディスク（HDD）などの大容量ストレージ・デバイスなどのインターフェース環境がこれまで以上に要求されます。また，高速なCPUも不可欠です。そこで，Windows XP活用に欠かせないDOS/V PCの自作パーツを選んでみました。

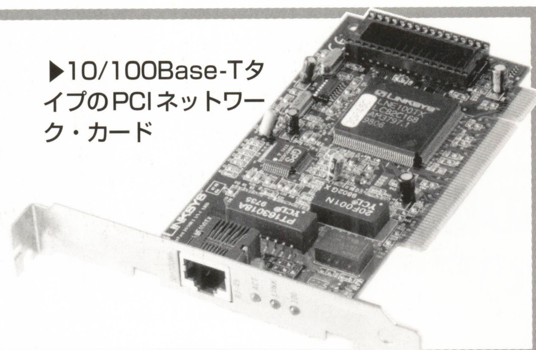
### グラフィック・カード

数え切れないほど多くの種類があるグラフィック・カードですが，予算が許す限り高速な3Dアクセラレーション機能を持つカードを選びましょう。RADEONやGeForce2MXシリーズなら3Dゲームも高速な描画画面で楽しめます。



### ネットワーク・カード

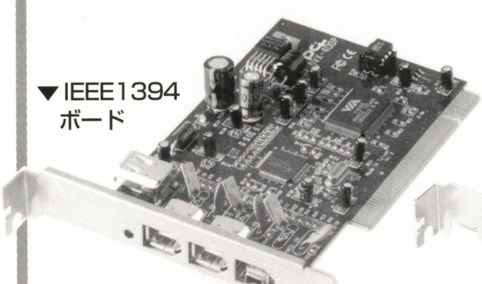
超高速通信が行えるブロードバンド回線やパソコン間のネット接続の利用には，ネットワーク・カードが欠かせません。「10/100 Base-T」タイプのPCIカードを選んでください。安価なものなら4,000円程度で購入可能なので，ほかのパーツと一緒に手に入れておきましょう。



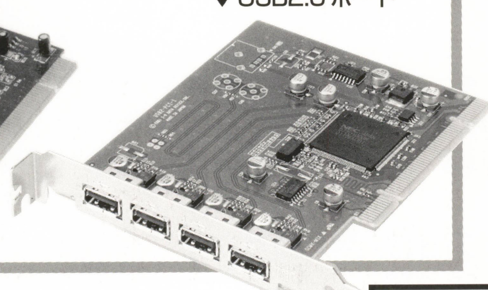
▶10/100Base-Tタイプの PCI ネットワーク・カード

### IEEE1394／USB2.0 インターフェース・カード

IEEE1394インターフェース・カードで，デジタルビデオ（DV）カメラで撮りためたDVテープのビデオ編集や外部ドライブの増設が行えます。予算に余裕があれば対応製品が増えている高速なデータ転送性能を持つUSB2.0カードもオススメ。



▼IEEE1394 ボード



▼USB2.0 ボード

### Ultra ATA/100 ボード＋HDD

手持ちのマザーボードを流用する場合，IDEインターフェースが，旧規格のUltra ATA/33やUltra ATA/66仕様のものもあります。そのようなときに便利に使えるのが，Ultra ATA/100対応のIDEインターフェース・カードです。秒速最大100Mbpsの高速転送が可能なUltra ATA/100規格で，大容量データを扱うデータベースやグラフィック編集などのソフトでも，スピーディな操作が行えるほか，Windows XPのムービーメーカーを使ったビデオ編集にもIDEカードの高速性能が生かせます。

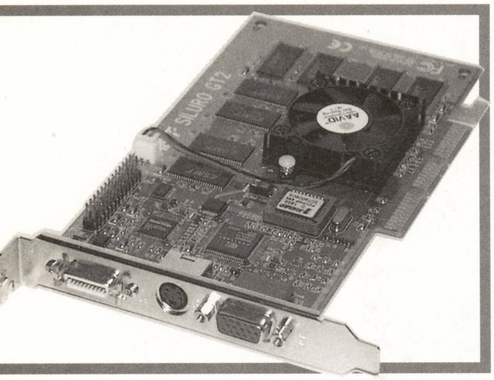



### CD-R／DVD-ROM ドライブ

Windows XP 標準のメディアプレーヤー XP の音楽CD作成やDVDビデオ再生機能の活用には，CD-RドライブやDVD-ROMドライブを選びましょう。予算が許せばCD-RとDVD-ROMの両方に対応したマルチドライブがベスト。

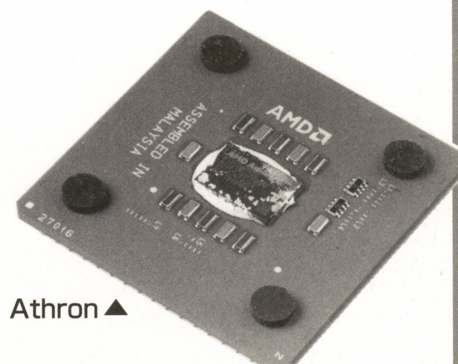
CD-R/RW 内蔵ドライブ▼



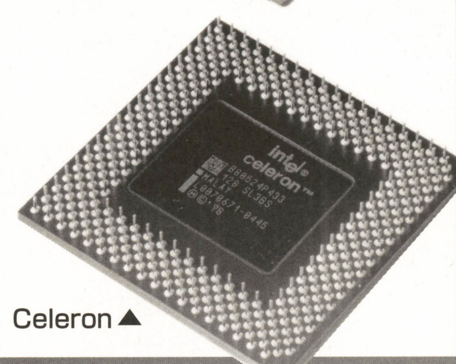


## CPU

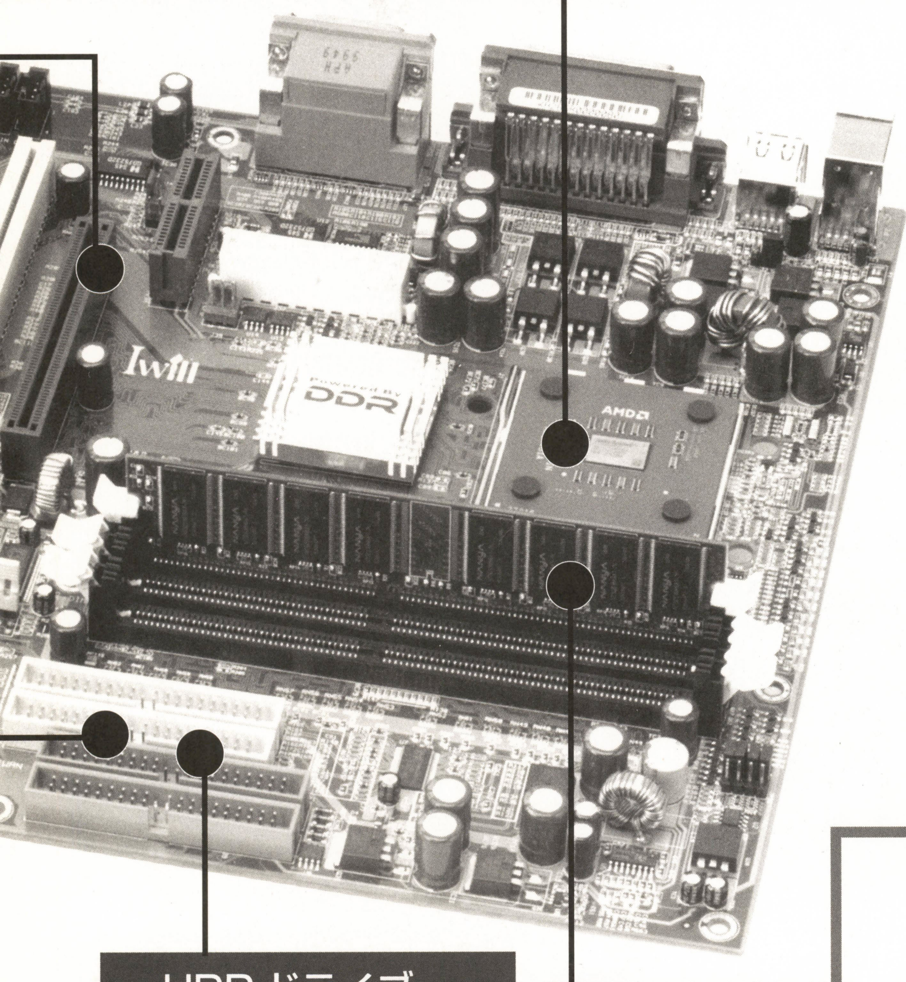
CPUの処理能力しだいで，パソコン全体の性能がほぼ決まってしまいます。それだけに，予算が許す限りクロック速度が高速なCPUを選びたいものです。現在，AMD社のAthronとDuron，インテル社のPentium Ⅲと4，Celeronがあり，コストパフォーマンスの点では，Athron／Duronが有利です。マザーボードが対応するCPUは，メーカーの違いやPentium Ⅲ／CeleronとPentium4では異なるので，購入時にはマザーボードと一緒にCPUを購入することが基本です。Windows XPでは，最低でも300MHzのCPUを推奨していますが，実用的ではありません。急激に価格が下がっていることもあり，1 GHz前後のCPUが良いでしょう。



Athron▲



Celeron▲



## HDDドライブ

最新マザーボードは，Ultra ATA/100規格のIDEインターフェースを採用。内蔵ハードディスクは，その規格に対応した10GB以上の製品を購入しよう。

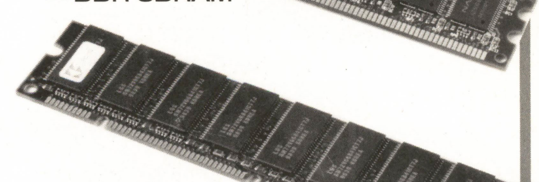


## メモリ・モジュール

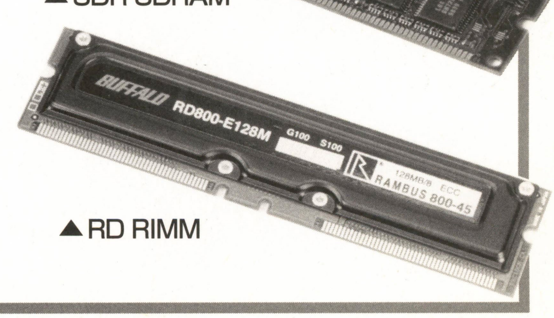
従来のSDR SDRAM（一般にはSDRAM）に加え，高速なデータ・リード/ライト性能を持つDDR SDRAMやRD RIMMなどがあり，マザーボードの仕様に合わせたタイプのメモリ・モジュールを選びます。メモリ容量はパソコン全体の処理スピードに影響するため，256MB以上がベスト。



▲DDR SDRAM

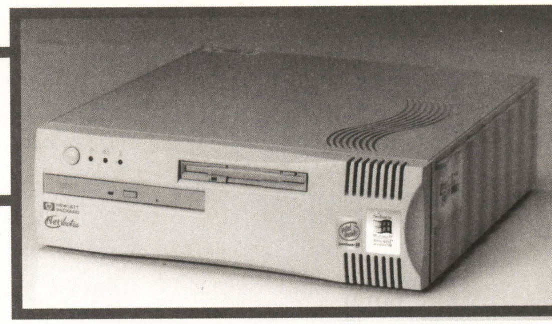


▲SDR SDRAM



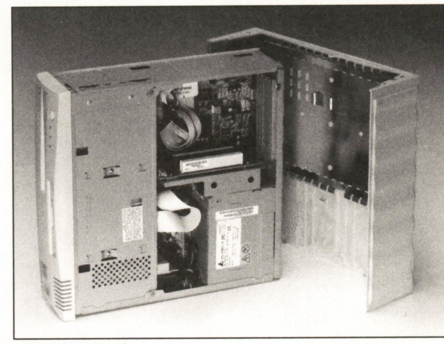
▲RD RIMM

# 手持ちのパソコンをXP向けにパワーアップしよう!!

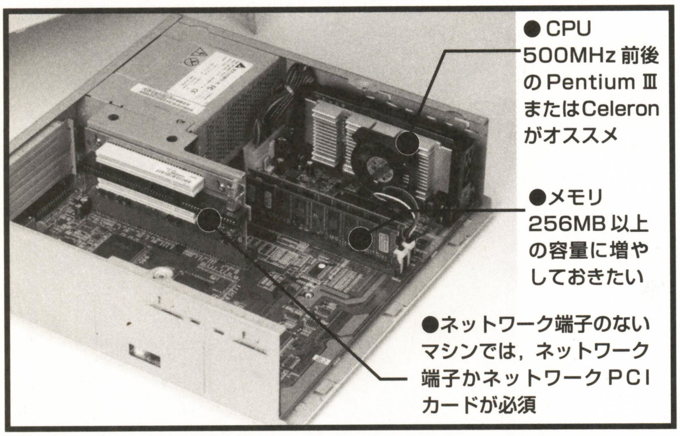


最新のマシンではないが，「手持ちのパソコンをそのままWindows XPでも使いたい」と考えている人も多いことでしょう。先のページで紹介したとおり，Winows XPのスペックによると，300MHz以上のCPUとメモリに128MBを推奨とあります。まず，Windows XPを導入する前に，このスペックを手持ちのパソコンが満たしているかをチェックしておきましょう。また，対応機種はDOS/V機のみとなっているのは，Windows Meや2000と同様です。PC-9800シリーズはサポートしていないので，注意してください。

推奨スペックぎりぎりのマシンでは，残念ながら動作が重く，実用には向かないようです。

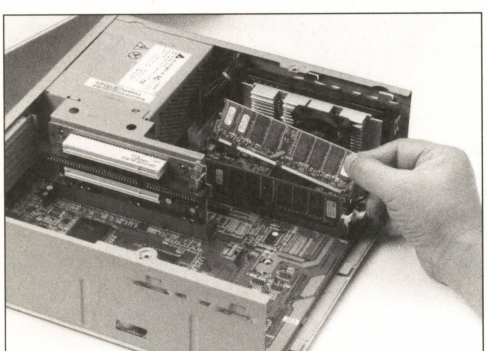


旧モデルでもCPUやメモリ容量がXPのスペックを満たせば，問題なくWindows XPが導入できる

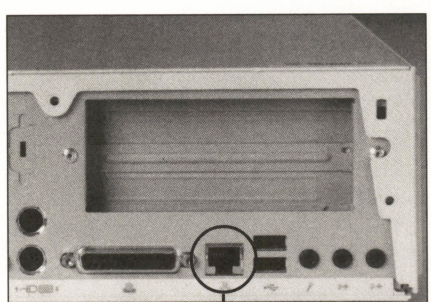



500MHz前後以上のPentium Ⅱ またはCeleron CPU搭載PCが最低レベルのスペックと考えてください。また，メモリは容量が多いほど，快適にWindows XPが動作します。メモリ価格が下がっていますので，思い切って256MB以上の容量に増やしましょう。日本ヒューレットパッカードの旧モデル，400MHz Pentium Ⅱ搭載デスクトップ機「NetVectra」に，標準メモリ64MBを256MBに増やしWindows XPを導入してみましたが，高速な画像処理を要求されるグラフィック・ソフトでは，少々もたつきを感じるものの，それ以外のソフトは問題ないレベルでした。

さらに，メモリの強化と同時にハードディスクも10GB程度の容量のものに交換しておけば万全です。



ほとんどのデスクトップ機では，メモリ・スロットが複数用意されているので，メモリを追加しておこう

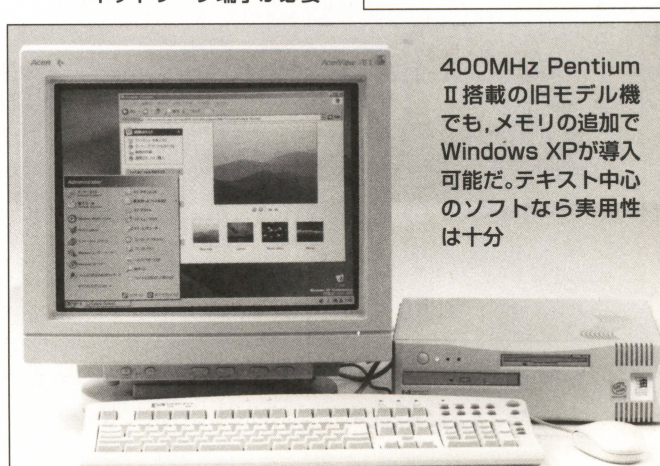


ブロードバンド接続には，ネットワーク端子が必要

記事協力：
コンピュータのおっと
http://www.ottonet.co.jp



大容量ハードディスクへの交換でビデオ編集やインターネットからの画像ダウンロードなども，余裕で楽しめる



400MHz Pentium Ⅱ搭載の旧モデル機でも，メモリの追加でWindows XPが導入可能だ。テキスト中心のソフトなら実用性は十分

# パーツセットの「キットPC」でXP対応マシンを低予算で導入!!



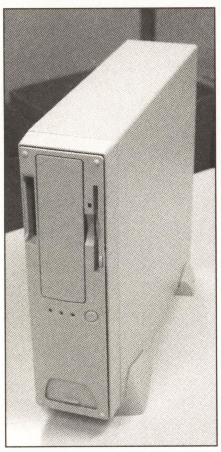

予算はぎりぎりでも，ワンランク上の性能を，という欲張りなユーザーに最適な製品が，「キットPC」です。

DOS／V機に必要なすべてのパーツがワンセットになっているため，面倒なパーツ選びなどは一切不要で，パーツの買い忘れもありません。なによりもうれしいのは，組み立て作業をユーザーみずからが行うことで，価格がきわめて安く設定されていることでしょう。キットPCは，自作の醍醐味と低価格という1台で二度楽しめるお得なDOS／Vマシンなのです。

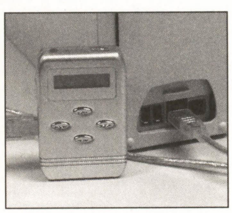
このページで紹介しているキットPC「SPLENDOR」シリーズは，若松通商（TEL 03-3257-0601　http://www.wakamatsu-net.com/）から発売の最新モデルで，いずれの機種もメモリ容量128MB以上，900MHz以上のCPUを搭載と，Windows XPの導入には十分な，充実したスペックが特徴です。価格も各パーツを集め自作するよりもグーンと安く，低予算でWindows XPを，という場合はもってこいでしょう。

スリムタイプ，オーディオ機器なみのデザインを持つコンパクトモデルなど，使用環境に合わせた機種選びができる点も見逃せません。

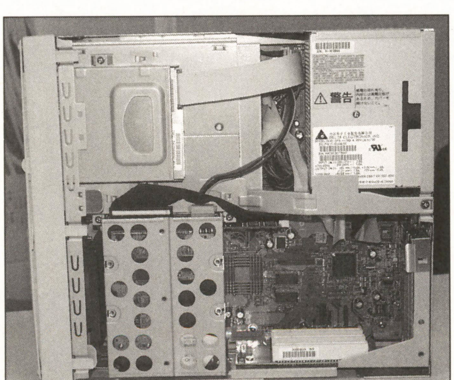
## 「SPLENDOR-FS1000」
価格69,800円

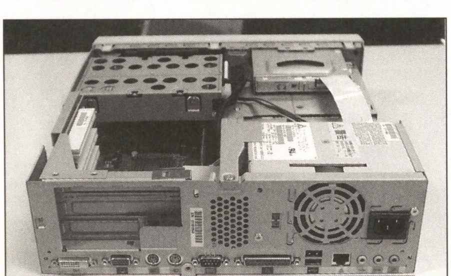


◀8倍速CD-R/RWドライブ搭載のスリム・タイプPC。PCカードスロットまである



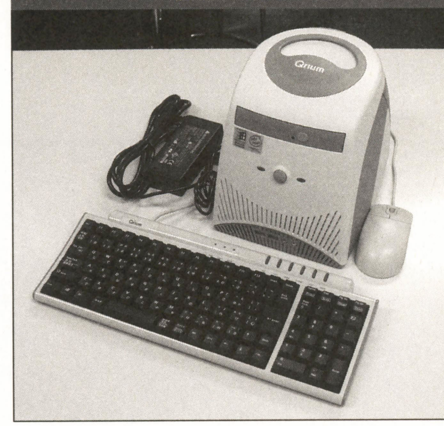
▲フロント・パネルにUSBとIEEE1394，デジタル音声出力端子を装備





▲CPUにはCeleron　1GHzを採用し，IEEE1394やネットワーク端子などWindows XPをフル活用できる多機能モデル

## 「Qrium CD1000」
価格49,800円

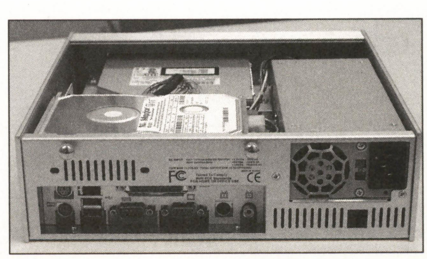


▲ユニークなキューブ・スタイルPC。シリアル・ポートなどを持たないレガシーフリー仕様。Celeron 1GHzを採用。Windows Me搭載

## 「SPLENDOR-MS900」
価格47,800円

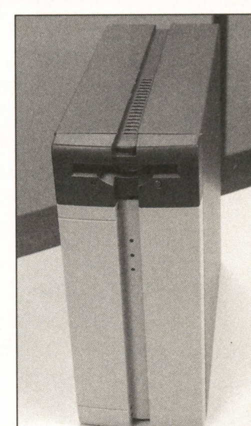


▲まるでオーディオ機器のようなスタイリッシュなデザインを採用したサイズ266×78×271mmの超コンパクト・モデル

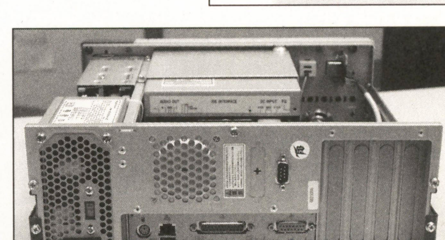


▲大型TVモニタへも画面表示が可能なコンポジット／S出力端子付き。CPUは，Celeron 900MHz

## 「SPLENDOR-GE1000」
価格39,800円



▶ツートンの落ち着いたカラーデザインを持ちビジネスPCとしてもオススメ。フロント・パネル内に，8倍速CD-R/RWドライブを装備



▲Celeron 1GHzを搭載し，ネットワーク端子を装備。ビジネス・ユースにも満足できるモデル

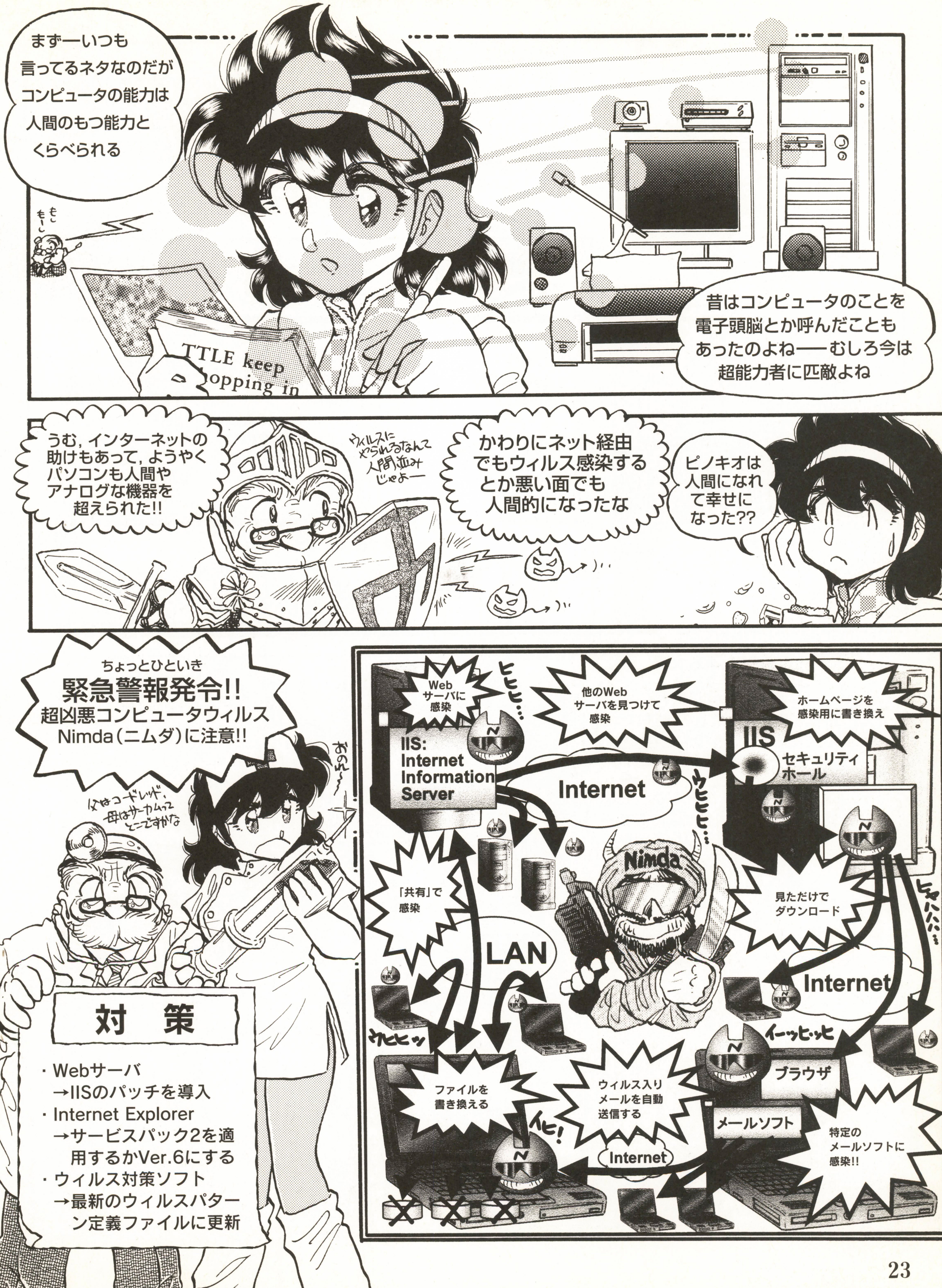
※各機種の価格は変動します。

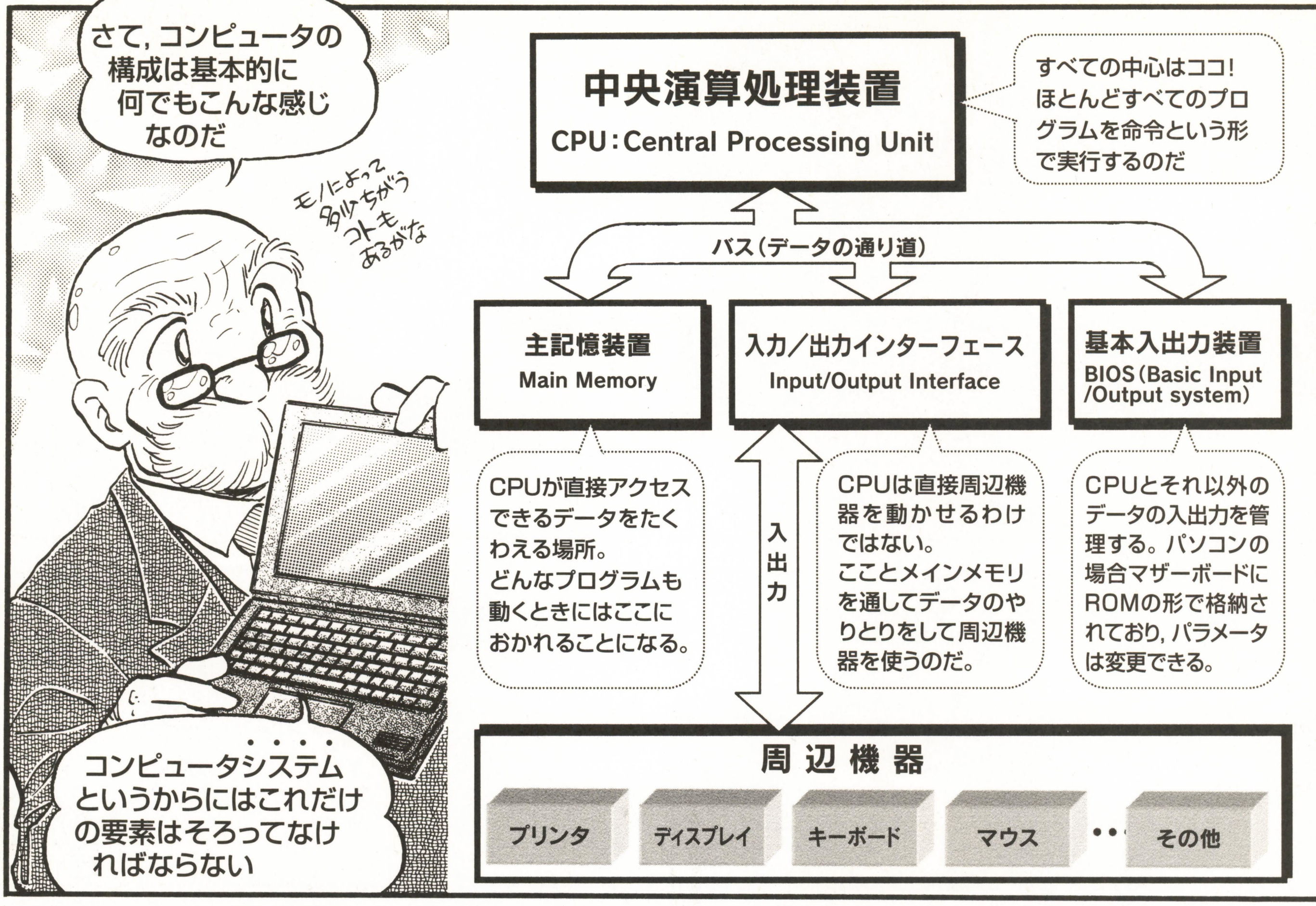
# パソコンレクチャー ザ・ビギニング

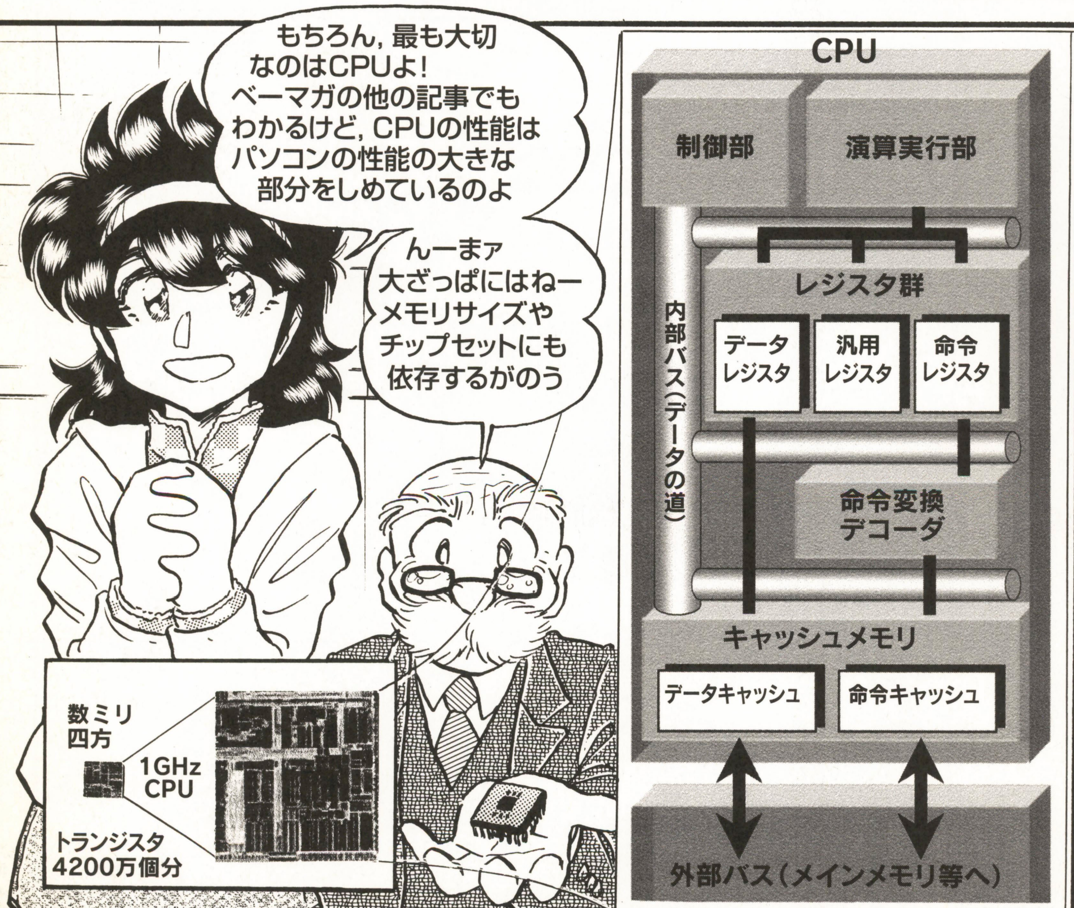
## Step4：「CPUって電子頭脳!?」

by くりひろし




まずーいつも
言ってるネタなのだが
コンピュータの能力は
人間のもつ能力と
くらべられる
もしもし
TTLE keep
opping in
昔はコンピュータのことを
電子頭脳とか呼んだことも
あったのよねーーむしろ今は
超能力者に匹敵よね
うむ, インターネットの
助けもあって, ようやく
パソコンも人間や
アナログな機器を
超えられた!!
ウィルスに
やられるなんて
人間並み
じゃよー
かわりにネット経由
でもウィルス感染する
とか悪い面でも
人間的になったな
ピノキオは
人間になれ
て幸せに
なった??
ちょっとひといき
緊急警報発令!!
超凶悪コンピュータウィルス
Nimda(ニムダ)に注意!!
おのれ〜〜
父はコードレッド、
母はサーカムって
とこですかな
対 策
・Webサーバ
→IISのパッチを導入
・Internet Explorer
→サービスパック2を適
用するかVer.6にする
・ウィルス対策ソフト
→最新のウィルスパター
ン定義ファイルに更新
Web
サーバに
感染
ヒヒヒ…
他のWeb
サーバを見つけて
感染
ホームページを
感染用に書き換え
IIS:
Internet
Information
Server
IIS
セキュリティ
ホール
Internet
ウヒヒヒ…
Nimda
「共有」で
感染
見ただけで
ダウンロード
ヒャハハハ…
LAN
Internet
ウヒヒッ
イーッヒッヒ
ブラウザ
ファイルを
書き換える
ウィルス入り
メールを自動
送信する
イヒ!
メールソフト
Internet
特定の
メールソフトに
感染!!


さて，コンピュータの構成は基本的に何でもこんな感じなのだ
モノによって多少ちがうコトもあるがな
中央演算処理装置
CPU：Central Processing Unit
すべての中心はココ！ほとんどすべてのプログラムを命令という形で実行するのだ
バス（データの通り道）
主記憶装置
Main Memory
入力／出力インターフェース
Input/Output Interface
基本入出力装置
BIOS（Basic Input /Output system）
CPUが直接アクセスできるデータをたくわえる場所。どんなプログラムも動くときにはここにおかれることになる。
入出力
CPUは直接周辺機器を動かせるわけではない。ここととメインメモリを通してデータのやりとりをして周辺機器を使うのだ。
CPUとそれ以外のデータの入出力を管理する。パソコンの場合マザーボードにROMの形で格納されており，パラメータは変更できる。
周辺機器
プリンタ
ディスプレイ
キーボード
マウス
・・・
その他
コンピュータ・シ・ス・テ・ムというからにはこれだけの要素はそろってなければならない


もちろん，最も大切なのはCPUよ！ベーマガの他の記事でもわかるけど，CPUの性能はパソコンの性能の大きな部分をしめているのよ
んーまァ大ざっぱにはねーメモリサイズやチップセットにも依存するがのう
CPU
制御部
演算実行部
内部バス（データの道）
レジスタ群
データレジスタ
汎用レジスタ
命令レジスタ
命令変換デコーダ
キャッシュメモリ
データキャッシュ
命令キャッシュ
外部バス（メインメモリ等へ）
数ミリ四方
1GHz CPU
トランジスタ4200万個分


それにしても…パソコンとゆーかCPUとゆーか…キカイが計算できる原理は前回まででわかったような気がしてたけど
じっさいのところ，どうやってコンピュータがプログラムを実行できるのかはまだナゾね


うむ！大変いい つーかもっとも な質問じゃ！
CやJavaのような「高級」プログラミング言語と コンピュータにしかわからん コトバ＝機械語（マシン語）は大ちがいだ
高級言語
A=1；
B=2；
C=A+B；
コンパイラ
アセンブリ言語
LD A, 1
LD B, 2
ADD A, B
LD C
※アセンブラを とばす場合も あります
アセンブラ
マシン語
71 72 58
4F 6F
01110001
01110010
01011000
01001110
実行可
ナルホド, いきなり0101… で書けるわけ ないもんねぇ アセンブラも…
作者とか むかしの人は アセンブリ言語で プログラムしたそーですが…
インタプリタは またちがうし
でかいプログラムも ちょっと ちがう
厳密には もっといろいろ あるのだが 基本はこんな もんだな
↑言語によってもちがうことがあります。
そっかあ こーやってプログラムを 動かすんだー。ちょっと 複雑なプログラムを 書いたらアッ…という間に 大量のマシン語に なるわよね
2000年
1GHz 4200万 トランジスタ
今や2GHzよねー
1990年
16MHz 70万 トランジスタ
1GHzのCPUは 1秒に10億回の 処理ステップを動かす ことができる…のも 今どきのパソコンなら わかるわァ
それにしても 輝ける21世紀最初 の年はまさに激動 の1年だったわねー
うむ, 21世紀は 混迷の中から新しい 世界を作りだすため のトレッキングな 世紀になろう
2001年7大デキゴト
①IT業界大不況リストラ
②2年遅れの恐怖の大王
③CPU2GHz時代に
④IMT2000サービス開始
⑤コイズミ流行する
⑥ゲーム業界再編
⑦インパク？
ははは っ 21世紀を明るく 楽しい世界にしていく のがボクら電脳界の 人々の役目なのさっ
え, また何か ヘンなこと言った…？
ギク
ほー――
めずらしく マトモなコトを…
そ, そりゃー このボクがその 気になれば不況 だろーがオサマ だろーが
おお
ま!!その分 給料もがっぱし 上げてもらわん となァハハハッ
プチッ
調子に乗るなあああぁぁあっっ
それでは 皆様良い お年を一…
ゴーン
ゴーン
DEMPA 電波新聞社
その年の除夜の 鐘は2001回 鳴ったそうな
Asisted by Eric, Special thanks to Mari.
To Be Continued…



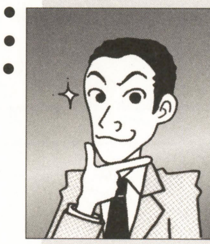
# はじめよう Microsoft ExcelでのBASICプログラミング

**協力** 富士通南九州システムエンジニアリング

## 第8回 コンピュータ対戦型のパズルゲームを作ろう

**この連載では，Microsoft Excelを使ってBASICのプログラミングを行います。今月はコンピュータとの対戦型ゲームをテーマに，簡単なゲームを作成しながらプログラミングの説明を行います。**

TEXT & PROGRAMMING 椛島たけし



うららかな日に，公園に出かけ，弁当を食べながらビールを飲む。ついでに木陰で昼寝。秋のちょっとしたぜいたく。時間がゆっくり流れ・・・，じつは猛スピードで流れていき，月曜になるのだった。

### コンピュータ対戦型のパズルゲーム

先月号では，「正解を求める」ゲームを作成しました。あらかじめ正解があるため，コンピュータに思考はありません。今回は，コンピュータ対戦型のパズルゲームを目標に，思考について考えてみることにしましょう。

### 対戦型パズルゲームの3タイプ

対戦型パズルゲームには大きくわけて三つのタイプがあります（第1図）。

#### ①ミスを待つタイプ

確実あるいは，ある程度はっきりとした勝ちパターンがあり，対戦者のミスを待つタイプのゲームです。このタイプには，三目ならべ，神経衰弱などが挙げられます。勝ちパターンが複雑なほど，つぎの②に近づきます。8月号掲載の「CUBEST」はこのタイプのゲームですね。

#### ②定石タイプ

確実な必勝法はありませんが，定石といわれる一般的なセオリーが存在するタイプのゲームです。連珠（五目ならべ），オセロ，囲碁，将棋，麻雀などが挙げられます。

#### ③確率に依存するもの

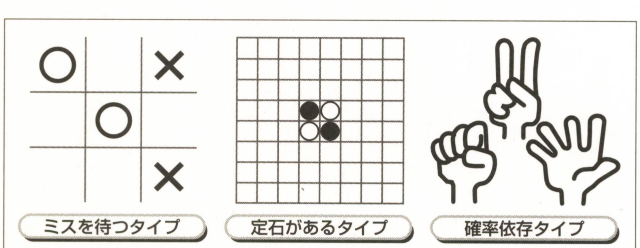
必勝法がなく，運（確率）に依存するタイプのゲームです。いわゆる，じゃんけん，コイントスなど確率的にイーブンなのが特徴です。

今回は，これらのタイプのなかで，③の確率に依存するタイプのゲームを作成してみます。確率的には，五分五分なものをいかに有利にするかがおもしろいところです。

### まずは，「じゃんけん」

確率に依存するゲームの最たるものは「じゃんけん」ですね。しかし，不思議と強い人や，弱い人がいるのも事実です。これは，クセによるところが大きいと思います。プログラミングでは，いかにこのクセを利用するかがポイントになります。

もちろん，インチキでもしない限り，コンピュータが100%勝つゲームを作成することはできません。長い目でみて「強い」ことを目指してコンピュータ対戦型のプログラムを作成することにしましょう。




《第1図》コンピュータ対戦型パズルゲームのタイプ

### ランダムなゲームをつくる

はじめに，「読み」の処理がない，完全にランダムなプログラムを作成します。じゃんけんでランダムとは，「グー」「チョキ」「パー」をバラバラに出すことですね。Excel VBAでランダムなデータを使用するにはRnd関数を使用します。Rnd関数を利用して三つのなかから，一つをランダムに選ぶプログラムはつぎのようになります。

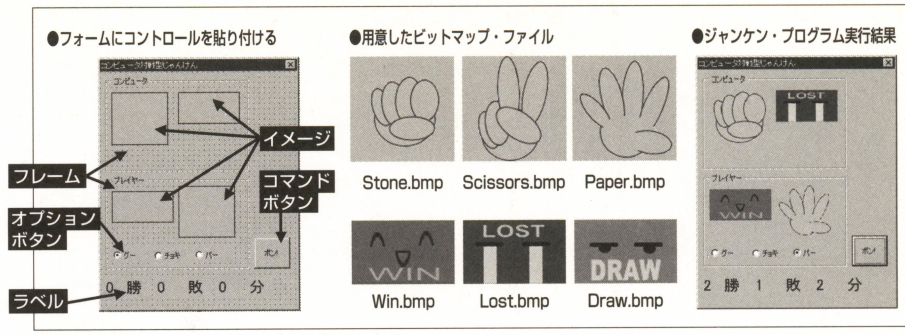
```
lngTempValue = Int((3 * Rnd) + 1)
select case lngTempValue
case 1
  処理1
case 2
  処理2
case 3
  処理3
end select
```

1をグー，2をチョキ，3をパーとすれば，簡単にランダムなじゃんけんプログラムを作成することができます。実際のプログラミング例を参考にしてください（第2図，リスト1）。

しかし，ランダムなジャンケンでは，プレイヤーが同じ手を出しつづけてもコンピュータが勝つ確率は1/3のままです。いわば，学習能力ゼロですね。

## 全体的に出す「手」の傾向による確率変動

コンピュータがグーで勝った場合，グーにポイントを与えます。負けたらポイントを失います（第3図）。それぞれの手で同じことを行います。勝った回数（確率）が多い手ほど，選ばれる確率が高いように，プログラミングします。




《第2図》ジャンケン・プログラムの例

（例）

```
    lngTempValue = Int((グーポイント + チョキポイント +パーポイント) * Rnd + 1)

    Select Case lngTempValue
    Case 1 To グーポイント
        lngPSSValueComp = 1 'グー
    Case グーポイント To グーポイント + チョキポイント
        lngPSSValueComp = 2 'チョキ
    Case チョキポイント To グーポイント + チョキポイント + パーポイント
        lngPSSValueComp = 3 'パー
    End Select
```

この方法を利用してロジックを変更したのが**リスト2**です（プロシージャのみ）。先ほどのリストの一部を置き換えて実行してみてください。

この方法では，コンピュータは「勝ったことのある手を多く出す」ようになります。プレイヤーが連続して同じ手を出すと，しだいにコンピュータが勝つようになります。この方法のマイナス面は，勝負の回数が少ない場合に差がでず，ランダムな場合と変わらなくなってしまうことです。

## 手のパターンによる分析

「グーを出して勝ったらつぎもグーを

《リスト1》コンピュータが出す手がランダムなジャンケン・プログラム

```
Option Explicit

Dim mlngWinCount  As Long  '通算勝ち星
Dim mlngLostCount  As Long '通算負け星
Dim mlngDrawCount  As Long '通算引き分け

Private Const PATH_BMP = "e:¥basicmagazine¥excel¥200112¥program¥"

'オプションボタンでパーが選ばれた時の処理
Private Sub optPaper_Click()
    If optPaper.Value = True Then
        imgPSSPlayer.Picture = LoadPicture(PATH_BMP & "Paper.bmp")
    End If
End Sub

'オプションボタンでチョキが選ばれた時の処理
Private Sub optScissors_Click()
    If optScissors.Value = True Then
        imgPSSPlayer.Picture = LoadPicture(PATH_BMP & "Scissors.bmp")
    End If
End Sub

'オプションボタンでグーが選ばれた時の処理
Private Sub optStone_Click()
    If optStone.Value = True Then
        imgPSSPlayer.Picture = LoadPicture(PATH_BMP & "Stone.bmp")
    End If

End Sub
'起動時の初期化処理
Private Sub UserForm_Initialize()
    'イメージデータの初期処理
    imgPSSPlayer.Picture = LoadPicture(PATH_BMP & "Stone.bmp")
    imgPSSComp.Picture = LoadPicture(PATH_BMP & "Stone.bmp")
    imgResultComp.Picture = LoadPicture(PATH_BMP & "Draw.bmp")
    imgResultPlayer.Picture = LoadPicture(PATH_BMP & "Draw.bmp")

End Sub

Private Sub cmdBattle_Click()
    Dim lngPSSValuePlayer As Long
    Dim lngResult As Long

    'オプションボタンの値をチェック
    If optStone.Value Then   'グー
        lngPSSValuePlayer = 1
    ElseIf optScissors Then  'チョキー
        lngPSSValuePlayer = 2
    ElseIf optPaper Then     'パー
        lngPSSValuePlayer = 3
    Else
        MsgBox "Error"
    End If

    '対戦処理
    Call procCompBattle1(lngPSSValuePlayer, lngResult)
    '結果表示処理
    Call procDispResult(lngResult)

End Sub
Private Sub procCompBattle1(lngPSSValuePlayer As Long, lngResult As Long)
    Dim lngPSSValueComp As Long
    'ランダムにコンピューターが出す手を決定
    lngPSSValueComp = Int((3 * Rnd) + 1)

    Select Case lngPSSValueComp
    Case 1 'コンピューターがグー
        imgPSSComp.Picture = LoadPicture(PATH_BMP & "Stone.bmp")
        Select Case lngPSSValuePlayer
        Case 1 'プレイヤーがグー
            lngResult = 0
        Case 2 'プレイヤーがチョキ
            lngResult = 1
        Case 3 'プレイヤーがパー
            lngResult = -1
        End Select
    Case 2 'コンピューターがチョキ
        imgPSSComp.Picture = LoadPicture(PATH_BMP & "Scissors.bmp")
        Select Case lngPSSValuePlayer
        Case 1 'プレイヤーがグー
            lngResult = -1
        Case 2 'プレイヤーがチョキ
            lngResult = 0
        Case 3 'プレイヤーがパー
            lngResult = 1
        End Select
    Case 3 'コンピューターがパー
        imgPSSComp.Picture = LoadPicture(PATH_BMP & "Paper.bmp")
        Select Case lngPSSValuePlayer
        Case 1 'プレイヤーがグー
            lngResult = 1
        Case 2 'プレイヤーがチョキ
            lngResult = -1
        Case 3 'プレイヤーがパー
            lngResult = 0
        End Select
    End Select
End Sub

'対戦結果表示処理
Private Sub procDispResult(lngResult As Long)

    Select Case lngResult
    Case -1  'コンピュータの負け
        imgResultComp.Picture = LoadPicture(PATH_BMP & "Lost.bmp")
        imgResultPlayer.Picture = LoadPicture(PATH_BMP & "Win.bmp")
        mlngWinCount = mlngWinCount + 1
    Case 0   'コンピュータの勝ち
        imgResultComp.Picture = LoadPicture(PATH_BMP & "Draw.bmp")
        imgResultPlayer.Picture = LoadPicture(PATH_BMP & "Draw.bmp")
        mlngDrawCount = mlngDrawCount + 1
    Case 1   'あいこ
        imgResultComp.Picture = LoadPicture(PATH_BMP & "Win.bmp")
        imgResultPlayer.Picture = LoadPicture(PATH_BMP & "Lost.bmp")
        mlngLostCount = mlngLostCount + 1
    Case Else
        MsgBox "Error"
    End Select

    '通算成績の表示
    lblWin.Caption = mlngWinCount
    lblLost.Caption = mlngLostCount
    lblDraw.Caption = mlngDrawCount
End Sub
```

《リスト2》出す手の傾向によるジャンケン・プログラム

```
'モジュールレベル変数追加
Dim mlngStonePoint  As Long   'グーのポイント
Dim mlngScissorsPoint As Long 'チョキのポイント
Dim mlngPaperPoint As Long    'パーのポイント

'置き換えるプロシージャ
Private Sub procCompBattle2(lngPSSValuePlayer As Long, lngResult As Long)
    Dim lngPSSValueComp As Long  'コンピュータの出す手
    Dim lngTempValue As Long     '

    '合計ポイントのなかからランダムな値を取り出す
    lngTempValue = Int((mlngStonePoint + mlngScissorsPoint + mlngPaperPoint) * Rnd +
1)

    '値の範囲により、手を決める
    Select Case lngTempValue
    Case 1 To mlngStonePoint
        lngPSSValueComp = 1
    Case mlngStonePoint To mlngStonePoint + mlngScissorsPoint
        lngPSSValueComp = 2
    Case mlngScissorsPoint To mlngStonePoint + mlngScissorsPoint + mlngPaperPoint
        lngPSSValueComp = 3
    Case Else
        MsgBox "err"
    End Select

    Select Case lngPSSValueComp 'コンピュータの手を判断
    Case 1 'コンピュータがグー
        'イメージ表示
        imgPSSComp.Picture = LoadPicture(PATH_BMP & "Stone.bmp")
        Select Case lngPSSValuePlayer '
        Case 1 'ユーザーがグー
            lngResult = 0
        Case 2 'ユーザーがチョキ
            lngResult = 1
        Case 3 'ユーザーがパー
            lngResult = -1
        End Select
        mlngStonePoint = mlngStonePoint + lngResult 'グーの累積ポイント計算

    Case 2 'コンピュータがチョキ
        'イメージ表示
        imgPSSComp.Picture = LoadPicture(PATH_BMP & "Scissors.bmp")
        Select Case lngPSSValuePlayer
        Case 1 'ユーザーがグー
            lngResult = -1
        Case 2 'ユーザーがチョキ
            lngResult = 0
        Case 3 'ユーザーがパー
            lngResult = 1
        End Select
        'チョキの累積ポイント計算
        mlngScissorsPoint = mlngScissorsPoint + lngResult

    Case 3 'コンピュータがパー
        'イメージ表示
        imgPSSComp.Picture = LoadPicture(PATH_BMP & "Paper.bmp")
        Select Case lngPSSValuePlayer
        Case 1 'ユーザーがグー
            lngResult = 1
        Case 2 'ユーザーがチョキ
            lngResult = -1
        Case 3 'ユーザーがパー
            lngResult = 0
        End Select
        'パーの累積ポイント計算
        mlngPaperPoint = mlngPaperPoint + lngResult
    Case Else
        MsgBox "Error"
    End Select


    '各累積ポイントの最小は1、最大は20とする
    If mlngStonePoint < 1 Then
        mlngStonePoint = 1
    End If
    If mlngStonePoint > 20 Then
        mlngStonePoint = 20
    End If

    If mlngScissorsPoint < 1 Then
        mlngScissorsPoint = 1
    End If
    If mlngScissorsPoint > 20 Then
        mlngScissorsPoint = 20
    End If

    If mlngPaperPoint < 1 Then
        mlngPaperPoint = 1
    End If
    If mlngPaperPoint > 20 Then
        mlngPaperPoint = 20
    End If

End Sub
```

出す」「負けたら手を変える」など，直前の手により，つぎに出す手を決める場合があります。先ほどのプログラムでは，直前の手はお構いなしでした。こんどは，直前の手を参考にするプログラミングを行ってみましょう。

まず，コンピュータとプレイヤーの出した手を記憶し，そのつぎにプレイヤーが何を出したかチェックします。この出した手をカウントしておきます。すると，前回のコンピュータとプレイヤーの手から，つぎにプレイヤーが何をよく出すのか知ることができます（第4図）。

リスト3に掲載のプロシージャに置き換えて試してみましょう。

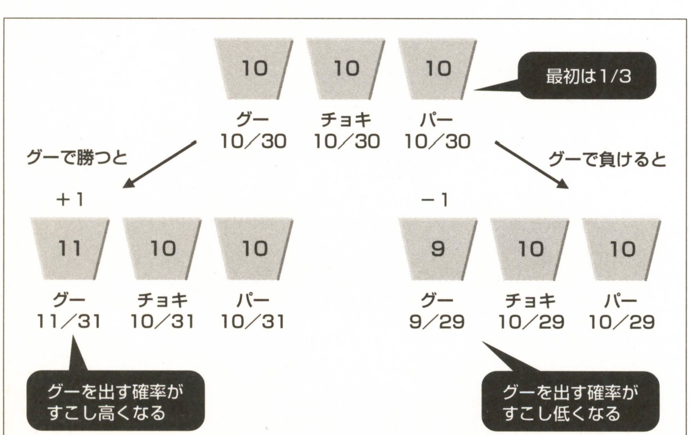
## コンピュータのクセをなくす

プレイヤーも，コンピュータがどのような手を出すか常に考えながら，出す手をかえてくるはずです。したがって，コンピュータのクセを読まれないようにすることも大事ですね。そのため，今まで紹介した方法をミックスすることにします。ミックスする場合もいろいろな方法があります。ちょっと挙げてみましょう。

- 前半・中盤・後半で戦法を変える
- ランダムに戦法を決める
- 最初にじゃんけん回数を決め，回数によって戦法を決める
- より確率が高い戦法を利用する

今回は，10回を一つのターンとして，ランダム→パターン→傾向と変化させてみることにします。リスト4を参考に，プロシージャを作り変えてください。最終的なリストは書面の都合で掲載できませんので，ベーマガのホームページからダウンロードしてください。

みなさんも，いろいろ応用して，より勝率の高いプログラムに発展させてみてください。




《第3図》勝ち負けによる確率変動

| 前回 | | 今回 | | |
|---|---|---|---|---|
| コンピュータ | プレイヤー | グー | チョキ | パー |
| グー | グー | 2 | 1 | 4 |
| チョキ | グー | 3 | 2 | 1 |
| パー | グー | 0 | 3 | 1 |
| グー | パー | 3 | 1 | 3 |
| チョキ | パー | 2 | 2 | 0 |
| パー | パー | 1 | 3 | 1 |
| グー | チョキ | 2 | 0 | 0 |
| チョキ | チョキ | 0 | 3 | 2 |
| パー | チョキ | 1 | 2 | 2 |

前回の結果から出した手をカウントアップしていく

(例)グーで負けた場合は，チョキを出す確率が高い

《第4図》パターンによる分析

## 《リスト3》手のパターン分析によるジャンケン・プログラム

```
'モジュールレベル変数追加
Dim lngPattern(3, 3, 3) As Long '前々回と前回の手の保存用配列

'置き換えるプロシージャ
Private Sub procCompBattle3(lngPSSValuePlayer As Long, lngResult As Long)
    Dim lngPSSValueComp As Long 'コンピュータの手
    Dim lngTempValue As Long
    Dim lngPSSValueCompTemp As Long
    Static lngPSSValueCompOLD As Long  '前回のコンピュータの手
    Static lngPSSValuePlayerOLD As Long '前回のプレイヤーの手

    '前回の手(コンピュータ、プレイヤー)による今回の手の決定
    'グーを出した回数とチョキを出した回数の比較
    If lngPattern(lngPSSValuePlayerOLD, lngPSSValueCompOLD, 1) > lngPattern(lngPSSVa
luePlayerOLD, lngPSSValueCompOLD, 2) Then
        'グーを出した回数とパーを出した回数の比較
        If lngPattern(lngPSSValuePlayerOLD, lngPSSValueCompOLD, 1) > lngPattern(lngP
SSValuePlayerOLD, lngPSSValueCompOLD, 3) Then
            lngPSSValueCompTemp = 3 'グーが多い場合パーを出す
        Else
            lngPSSValueCompTemp = 2 'パーが多い場合、チョキを出す
        End If
    'チョキを出した回数とパーを出した回数の比較
    ElseIf lngPattern(lngPSSValuePlayerOLD, lngPSSValueCompOLD, 2) > lngPattern(lngP
SSValuePlayerOLD, lngPSSValueCompOLD, 3) Then
        lngPSSValueCompTemp = 1 'チョキが多い場合、パーを出す
    Else
        lngPSSValueCompTemp = 2 'パーが多い場合、チョキを出す
    End If

    lngPSSValueComp = lngPSSValueCompTemp

    Select Case lngPSSValueComp
    Case 1 'コンピュータがグー
        imgPSSComp.Picture = LoadPicture(PATH_BMP & "Stone.bmp")
        Select Case lngPSSValuePlayer
        Case 1 'ユーザーがグー
            lngResult = 0
        Case 2 'ユーザーがチョキ
            lngResult = 1
        Case 3 'ユーザーがパー
            lngResult = -1
        End Select

    Case 2 'コンピュータがチョキ
        imgPSSComp.Picture = LoadPicture(PATH_BMP & "Scissors.bmp")
        Select Case lngPSSValuePlayer
        Case 1 'ユーザーがグー
            lngResult = -1
        Case 2 'ユーザーがチョキ
            lngResult = 0
        Case 3 'ユーザーがパー
            lngResult = 1
        End Select

    Case 3 'コンピュータがパー
        imgPSSComp.Picture = LoadPicture(PATH_BMP & "Paper.bmp")
        Select Case lngPSSValuePlayer
        Case 1 'ユーザーがグー
            lngResult = 1
        Case 2 'ユーザーがチョキ
            lngResult = -1
        Case 3 'ユーザーがパー
            lngResult = 0
        End Select
    Case Else
        MsgBox "error"
    End Select

    '前回のコンピュータとプレイヤーの手と今回のプレイヤーの手の組み合わせを
    'カウントアップ
    lngPattern(lngPSSValuePlayerOLD, lngPSSValueCompOLD, lngPSSValuePlayer) = _
        lngPattern(lngPSSValuePlayerOLD, lngPSSValueCompOLD, lngPSSValuePlayer)
+ 1

    '今回のコンピュータとプレイヤーの手を保存する
    lngPSSValueCompOLD = lngPSSValueComp
    lngPSSValuePlayerOLD = lngPSSValuePlayer

End Sub
```

## 《第1表》コントロールのプロパティ設定

| | |
|---|---|
| コントロール | フォーム |
| オブジェクト名 | UserForm1 |
| Caption | コンピュータ対戦型じゃんけん |
| コントロール | イメージ |
| オブジェクト名 | imgPSSComp |
| PictureSizeMode | 1-fmPictureSizeModeStretch |
| コントロール | イメージ |
| オブジェクト名 | imgPSSPlayer |
| PictureSizeMode | 1-fmPictureSizeModeStretch |
| コントロール | イメージ |
| オブジェクト名 | imgResultComp |
| PictureSizeMode | 1-fmPictureSizeModeStretch |
| コントロール | イメージ |
| オブジェクト名 | imgResultPlayer |
| PictureSizeMode | 1-fmPictureSizeModeStretch |
| コントロール | コマンドボタン |
| オブジェクト名 | cmdBattle |
| Caption | ポン |
| コントロール | オプションボタン |
| オブジェクト名 | optStone |
| Value | True |
| コントロール | オプションボタン |
| オブジェクト名 | optScissors |
| Value | False |
| コントロール | オプションボタン |
| オブジェクト名 | optPaper |
| Value | False |
| コントロール | ラベル |
| オブジェクト名 | lblWin |
| コントロール | ラベル |
| オブジェクト名 | lblLost |
| コントロール | ラベル |
| オブジェクト名 | lblDraw |

## 《リスト4》ランダム→パターン→傾向と変化させたジャンケン・プログラム

```
'ミックスした例(一部)

    Static lngBattleCount As Long

    lngBattleCount = lngBattleCount + 1

    Select Case lngBattleCount Mod 10
    Case 1 To 3  'ランダム
        lngPSSValueComp = Int((3 * Rnd) + 1)

    Case 4 To 7
        '前回の手(コンピュータ、プレイヤー)による今回の手の決定
        'グーを出した回数とチョキを出した回数の比較
        If lngPattern(lngPSSValuePlayerOLD, lngPSSValueCompOLD, 1) > lngPattern(lngPSSValuePlayerOLD, lngPSSValue
CompOLD, 2) Then
            'グーを出した回数とパーを出した回数の比較
            If lngPattern(lngPSSValuePlayerOLD, lngPSSValueCompOLD, 1) > lngPattern(lngPSSValuePlayerOLD, lngPSSV
alueCompOLD, 3) Then
                lngPSSValueCompTemp = 3 'グーが多い場合パーを出す
            Else
                lngPSSValueCompTemp = 2 'パーが多い場合、チョキを出す
            End If
        'チョキを出した回数とパーを出した回数の比較
        ElseIf lngPattern(lngPSSValuePlayerOLD, lngPSSValueCompOLD, 2) > lngPattern(lngPSSValuePlayerOLD, lngPSSV
alueCompOLD, 3) Then
            lngPSSValueCompTemp = 1 'チョキが多い場合、パーを出す
        Else
            lngPSSValueCompTemp = 2 'パーが多い場合、チョキを出す
        End If
        lngPSSValueComp = lngPSSValueCompTemp

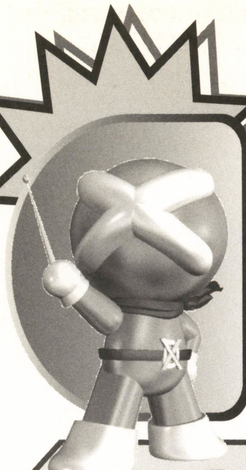
    Case Else
        lngTempValue = Int((mlngStonePoint + mlngScissorsPoint + mlngPaperPoint) * Rnd + 1)
        Select Case lngTempValue
        Case 1 To mlngStonePoint
            lngPSSValueComp = 1
        Case mlngStonePoint To mlngStonePoint + mlngScissorsPoint
            lngPSSValueComp = 2
        Case mlngScissorsPoint To mlngStonePoint + mlngScissorsPoint + mlngPaperPoint
            lngPSSValueComp = 3
        End Select
    End Select

'______________
```

## Low Techinal Column 「わたしのジャンケン必勝法」

筆者の会社では宴会のとき，「にぎりジャン」というものが行われる。参加者全員が左手にコインを握りしめ，隣の人とじゃんけんをする。勝った人はコインをもらう。これを繰り返し，最後に勝ちのこった人が参加者分のコインを受け取れるといったものだ（コインは決して百円玉などではありません）。単純だが，これが結構盛り上がるのだ。

このような，人数が多いじゃんけんほど「チョキ」が勝つ確率が高い(持論)。筆者は最初から最後までチョキを出して勝ち抜き，優勝したことがある。とくに，3人以上でいっしょにじゃんけんをするときは，最初から最後まで「チョキ」だ。ただし，気合が入ると「グー」を連発する体育会系がときどきいるので要注意である。



▲X君

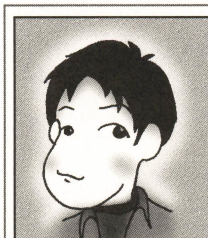
マイクロソフトの技術者がDirectXテクノロジーを伝授する！

# IE+DirectAnimation マルチメディア・プログラミング

## 第9回　キー・イベントの処理テクニックを紹介！

**このコーナーでは，『Microsoft Internet Explorer（以下IEと略）』と『DirectAnimation[*1]』を使って，誰でも簡単に作れるマルチメディア・プログラムを紹介します。今回は，「キーボードのイベントにアニメーションを反応させる」というテクニックを取りあげます。また，差し替え可能なオブジェクトという新しい概念についても紹介しましょう。**


TEXT&PROGRAMING●平井昌人



横浜市生まれの&H25歳男性。1993年4月に，開発言語製品のサポート・エンジニアとしてマイクロソフトに入社。現在はシステム・エンジニア（SE）としてVisual Studioを中心とした開発系の啓蒙・支援活動を行っている。先日，数年ぶりに徹夜マージャンをした。大勝利を収めたものの，体調がもとにもどるまで3日もかかった。体の衰えを感じる今日このごろ。


### 差し替え可能なオブジェクトとは？

はじめに，DirectAnimationの「差し替え可能なオブジェクト」という新しい概念について解説します。

差し替え可能なオブジェクトとは，好きなときに好きな内容に切り替えることが可能な，DirectAnimationのオブジェクトです。実例としては，

- ●ボタンを押したときに，イメージ全体を別のまったくちがったイメージに変える
- ●クリックしたら図形の形や文字を変える
- ●スライダーコントロールでサウンドのボリュームを変化させる

などがあります。

とくに，マウスやキーボード，ボタンコントロールやスライダーコントロールなどのイベントプロシージャから，実行中のDirectAnimationオブジェクトの内容を切り替えたいときに使います。

DirectAnimationが提供するオブジェクトのほとんどは，差し替え可能なオブジェクトとして利用することができます。とくにDAImage（イメージ），DASound（サウンド），DAColor（色），DANumber(数値)，DAString（文字列）がよく使われます。

差し替え可能なオブジェクトを利用する場合，通常の処理の他に，つぎの二つのステップが必要になります。

- ●差し替え可能なオブジェクトを定義する
- ●オブジェクトを差し替える

### 差し替え可能なオブジェクトの定義～ModifiableBehaviorメソッド

差し替え可能なオブジェクトを定義する場合は，DAStatics オブジェクトのModifiable Behaviorメソッド[*2]を使います。

メソッドの引数に，オブジェクトの初期値が必要です。ModifiableBehaviorメソッドの構文は，つぎのようになります。

```
Set obj = m.ModifiableBehavior(初期オブジェクト)
```

つぎに，差し替え可能なオブジェクトを定義する例をいくつかあげておきます。

**●例1：差し替え可能なDAColorオブジェクト**

```
Set col = m.ModifiableBehavior(m.Red)
```

**●例2：差し替え可能なDAImageオブジェクト**

```
Set defimg = m.ImportImage("A.GIF")
Set img = m.ModifiableBehavior
(defimg)
```

**●例3：差し替え可能なDASoundオブジェクト**

```
Set defsnd = m.ImportSound
("X.WAV").Sound
Set snd = m.ModifiableBehavior
```

＊1　DirectAnimation
2Dや3Dグラフィックス，サウンド，映像などを総合的にサポートする高レベルなAPIを提供している。IEに統合されており，インターネットでの利用を強く意識した設計となっている。リファレンスは，http://www.microsoft.com/DIRECTX/dxm/ を参照。

＊2　ModifiableBehaviorメソッド
差し替え可能なDirectAnimationのオブジェクトを定義するメソッド。引数で指定したDirectAnimationのオブジェクトが初期値になり，あたえたオブジェクトと同じ型のオブジェクトで受け取らなければならない。実行中にオブジェクトを差し替えることができる以外は，通常のDirectAnimationのオブジェクトと同じように扱うことができる。

```
(defsnd)
```

●例4：差し替え可能なDAPath2オブジェクト

```
Set defPth = m.Oval(32, 32)
Set pth = m.ModifiableBehavior
(defPth)
```

## 差し替え可能なDANumberオブジェクトの定義 ～ModifiableNumberメソッド

差し替え可能なオブジェクトがDANumberの場合，DAStatics オブジェクトのModifiable Numberメソッドを使うと便利です。初期値に，オブジェクトでなく数値リテラル*3や数値変数を指定できます。

ModifiableNumberメソッドの構文は，つぎのようになります。

```
Set num = m.ModifiableNumber(数値)
```

●例5：差し替え可能なDANumberオブジェクト

```
Set num = m.ModifiableNumber(1.5)
```

このコードは，ModifiableBehaviorメソッドを使ってつぎのように記述してもOKです。

```
Set defNum = m.DANumber(1.5)
Set num = m.ModifiableBehavior
(defNum)
```

## 差し替え可能なDAStringオブジェクトの定義 ～ModifiableStringメソッド

差し替え可能なオブジェクトがDAStringの場合は，DAStaticsオブジェクトのModifiable Stringメソッドを使うこともできます。ModifiableStringメソッドも，初期値に文字リテラル*4や文字列変数が使えます。

ModifiableStringメソッドの構文は，つぎのようになります。

```
Set moj = m.ModifiableString
("文字列")
```

●例6：差し替え可能なDAStringオブジェクト

```
Set abc = m.ModifiableString("ABC")
```

上のコードは，ModifiableBehaviorメソッドを使ってつぎのように記述してもOKです。

```
Set newAbc = m.DAString("ABC")
Set abc = m.ModifiableBehavior
(newAbc)
```

## オブジェクトの差し替え ～SwitchToメソッド

ModifiableBehaviorメソッドで作成した差し替え可能なオブジェクトは，いつでも好きなときにSwitchToメソッドで別の内容に差し替えることができます。

構文は，つぎのようになります。

```
obj.SwitchTo オブジェクト
```

つぎの例は，現在のイメージを新しい「青のベタ塗りイメージ」(newImg) に差し替えます。

```
Set newImg = m.SolidColorImage
(m.Blue)
img.SwitchTo newImg
```

また，変数を使わずに，つぎのように直接オブジェクトを渡すこともできます。

```
img.SwitchTo m.SolidColorImage
(m.Blue)
```

## DANumberオブジェクトの差し替え ～SwitchToNumberメソッド

DANumberオブジェクトの差し替えは，SwitchToNumberメソッドを使うと便利です。SwitchToメソッドとはちがい，数値リテラルや数値変数 (Double型) で直接差し替えることができます。

構文は，つぎのようになります。

```
num.SwitchToNumber 数値
```

つぎの例は，DANumberオブジェクトの中身を新しい値0.7に差し替えます。

```
num.SwitchToNumber 0.7
```

変動値を表すDANumberオブジェクトへ差し替える場合は，SwitchToメソッドのほうを使います。つぎの例は，0～1まで2秒で変位するDANumberオブジェクトに差し替えます。

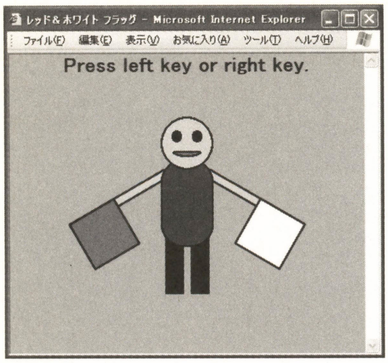
```
Set newNum = m.Interpolate(0, 1, 2)
num.SwitchTo newNum
```

## DAStringオブジェクトの差し替え ～SwitchToStringメソッド

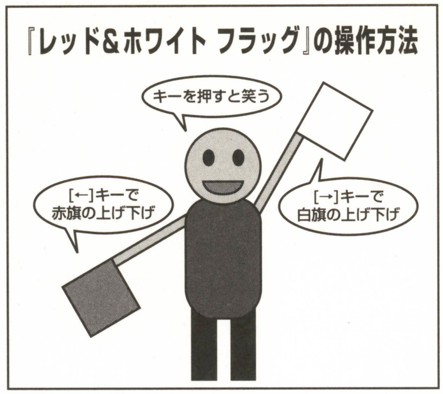
DAStringオブジェクトの差し替えは，文字

*3　数値リテラル
「n = 123」の123のように，コード中に直接，数値を記述している部分を，数値リテラルと呼ぶ。

*4　文字リテラル
「s = "ABC"」の"ABC"のようにコード中に直接，文字列を記述している部分を文字リテラルと呼ぶ。




《写真1》『レッド＆ホワイト フラッグ』の実行画。[←][→]キーを押してみよう




《第1図》二つのキーで，旗の上げ下げを行う。細かい動きに注目してほしい！

リテラルや文字列変数を直接あたえることができるSwitchToStringメソッドが便利です。

構文は，つぎのようになります。

```
abc.SwitchToString "文字列"
```

つぎの例は，DAStringオブジェクトの中身を新しい文字列“XYZ”に差し替えます。

```
abc.SwitchToString "XYZ"
```

## キーボードのイベントを利用する方法

つぎに，キーイベント*5の取得方法について解説します。DirectAnimationでキーボードのイベントを利用する方法は，つぎの二つのパターンがあります。

- **イベントプロシージャを使う**
- **DAEventオブジェクトを使う**

VBのような一般的なWindowsのプログラミングでは，イベントプロシージャを使ってキーボードのイベントをハンドリングします。DirectAnimationでは，他にあらかじめアニメーションの設定にイベント処理を組み込んでおくという独自の方法も提供しています。

作成するアニメーションのスタイルによって，この二つを使いわけてください。

## DAEventオブジェクトを利用したキーイベントの処理

DirectAnimationでは，イベントを処理するためのDAEventオブジェクトが提供されています。はじめに，DAEventオブジェクトを使って，あらかじめアニメーションにイベントを仕込んでおく方法を取り上げましょう。

ここでは，簡単なサンプル・プログラム『レッド＆ホワイト フラッグ』をもとに解説していきます。ソースコードは，**リスト1**です。このサンプルを実行すると，**写真1**のように中央に赤旗と白旗を持った人形が現れます。操作方法は簡単で，**第1図**のとおりです。

- **[←]方向キー：赤旗の上げ下げ**
- **[→]方向キー：白旗の上げ下げ**

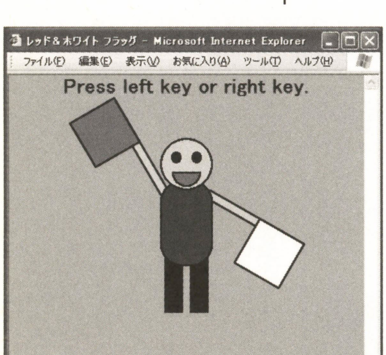
**写真2**は，[←]キーを押して赤旗を上げた状態です。旗を上げ下げするときに，人形の唇が動いてニコッと笑います。さらに，[←]キーと[→]キーを同時に押して，両旗を一度に動かすこともできます。

## 人形は基本的な図形のパーツを合成する！

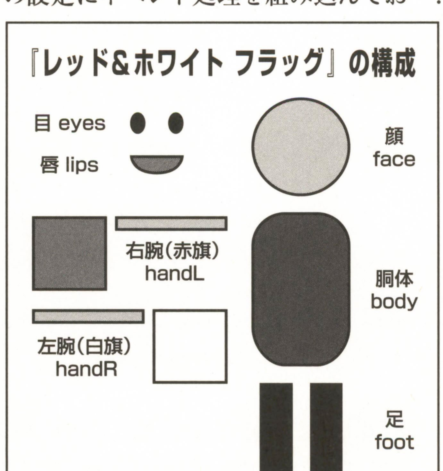
リストを見ると，長々とコーディングしてありますね。これは，画像ファイルを使わず，**第2図**のように人形のパーツをすべてDirectAnimationで描画しているためです。

この旗を持った人形のパーツの作成で，おもしろいテクニックを使っています。それは，白旗のイメージの作成です。**第3図**のように，白旗を赤旗と同じ方向，同じ向きに作成しておき，最後にScale2メソッドで左右ひっくり返すというテクニックです。

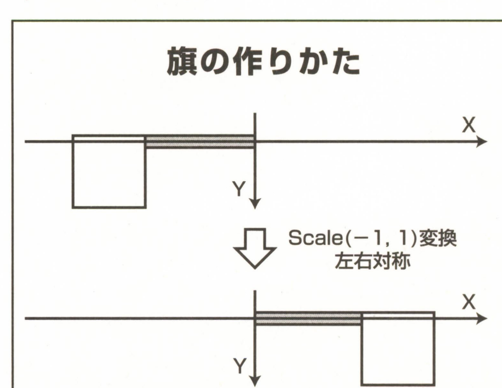
***5 キーイベント**
DirectAnimationでのキーイベントは，コントロール（DAControl）にフォーカスがあるときでないと発生しない。サンプルを実行してもまったくキーが効かない場合，フォーカスを失っている。一度マウスでIEの画面をクリックしてみるとよい。サンプルでは，スタート時に「DAControl.Focus」のFocusメソッドを使って強制的にコントロールにフォーカスを移動させているが，実行中に別のウィンドウへ移動したり，フォーカス移動キー（TAB）などを押してしまうと，コントロールからフォーカスが外れてしまうので注意。



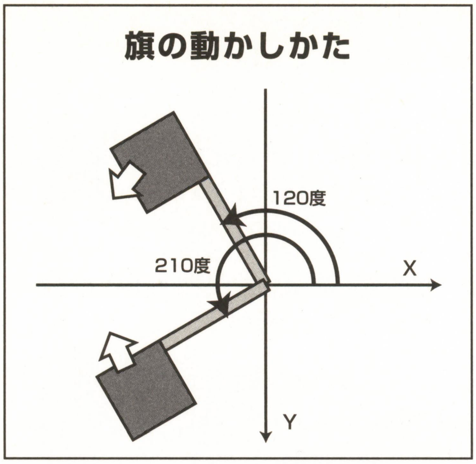

《写真2》『レッド＆ホワイト フラッグ』は，よく見ると旗上げゲームか？



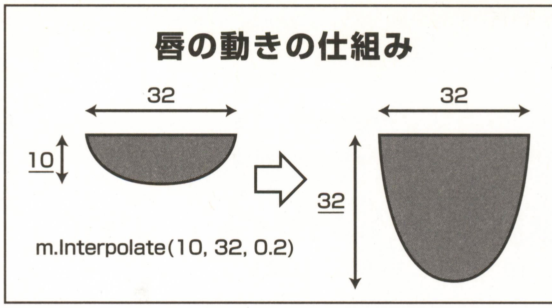

《第2図》描画しているイメージのパーツは，このように構成されているのだ




《第3図》白旗は，赤旗と同じ作りかたで，最後に左右反転しているところがポイントだ




《第4図》旗の上げ下げは，回転変換を利用している




《第5図》半円の高さを変化させて，唇の動きを実現している

つぎのように，X軸方向に−1倍すれば，左右反転するわけです。

```
Set handR = handR.Transform (m.Scale2(-1, 1))
```

## 旗の上げ下げは回転変換を利用

旗の上げ下げは，**第4図**のように回転変換を行うRotate2Animメソッドにイベントを組み込んで，変動する回転角（roteR）をあたえて制御しています。

```
Set handR = handR.Transform
(m.Rotate2Anim(roteR))
```

なお，回転角の変動値はInterpolateメソッドを使っているので，上げていく／下げていく動きがなめらかにアニメーションします。

## 唇の動きは半円の高さを変化

唇の動きは，**第5図**のように扇形（サンプルでは半円）の軌跡を定義するPieRadiansAnimメソッドの高さパラメータに，イベントを組み込んだ変動値（numY）をあたえることで制御しています。

numYもInterpolateメソッドを使って変位させているので，唇が開く／閉じるとう動きがなめらかにアニメーションします。

## DirectAnimationが提供しているキーボードのイベント

キーイベントを処理するために，つぎの二つのメソッドで用意されています。どちらも，DAEventオブジェクトを返します。

●m.KeyDown(KeyCode)：KeyCodeで指定されるキーが押された

●m.KeyUp(KeyCode)：KeyCodeで指定されるキーが離された

## キーイベントのDAEventオブジェクトの定義～KeyDown／KeyUpメソッド

DAStaticsオブジェクトのKeyDownメソッドやKeyUpメソッド*6を使って，取得したいキーイベントを定義します。

KeyDownメソッド／KeyUpメソッドの引数には，取得したいキーのキーコードを指定します。**第1表**に，KeyDownメソッド／KeyUpメソッドに指定するキーコードを示してあります。原則として，対象のキーのアスキー・コードに，16進数*7の&H10000を加算した値になります。

つぎのコードは，キーコード&H10025

**＊6　KeyDownメソッド／KeyUpメソッド**
DirectAnimationのKeyDownメソッドおよびKeyUpメソッドで取得できるのは，方向キーやファンクション・キーなどの特殊キーのみである。残念ながら，[A]～[Z]のような一般キーは，このメソッドで取得できない。

**＊7　16進数**
1桁が0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,Fの15で表現される数。16で位があがる。VBScriptでは，数値の先頭に"&H"を付けると16進表記になる。"&HFF"は16進数でFFを意味し，10進数に直すと255である。

《第1表》KeyDown／KeyUpで指定するキーコード

| キー | KeyCode（16進） | キー | KeyCode(16進) |
|---|---|---|---|
| PGUP | 10021 | F1 | 10070 |
| PGDN | 10022 | F2 | 10071 |
| END | 10023 | F3 | 10072 |
| HOME | 10024 | F4 | 10073 |
| LEFT | 10025 | F5 | 10074 |
| UP | 10026 | F6 | 10075 |
| RIGHT | 10027 | F7 | 10076 |
| DOWN | 10028 | F8 | 10077 |
| INSERT | 1002D | F9 | 10078 |
| DELETE | 1002E | F10 | 10079 |
| NUMPAD0 | 10060 | F11 | 1007A |
| NUMPAD1 | 10061 | F12 | 1007B |
| NUMPAD2 | 10062 | F13 | 1007C |
| NUMPAD3 | 10063 | F14 | 1007D |
| NUMPAD4 | 10064 | F15 | 1007E |
| NUMPAD5 | 10065 | F16 | 1007F |
| NUMPAD6 | 10066 | F17 | 10080 |
| NUMPAD7 | 10067 | F18 | 10081 |
| NUMPAD8 | 10068 | F19 | 10082 |
| NUMPAD9 | 10069 | F20 | 10083 |
| MULTIPLY | 1006A | F21 | 10084 |
| ADD | 1006B | F22 | 10085 |
| SEPARATOR | 1006C | F23 | 10086 |
| SUBTRACT | 1006D | F24 | 10087 |
| DECIMAL | 1006E | | |

**＊8　インスタンス化**
インスタンスは，オブジェクトがメモリ上に確保され実態となっていることをいう。インスタンス化するとは，そのオブジェクトを空っぽの状態で生成するということ。たとえば，DirectAnimation.DAColorをCreateObjectメソッドでインスタンス化すると，色がまだ定義されていない未初期化のDAColorオブジェクトが作成できる。Initメソッドは，このような初期化されていないオブジェクトに対して一度だけ使うことができる。

**＊9　AndEventメソッド**
AndEventメソッドは，「AかつB」でイベント（DAEventオブジェクト）同士の論理積を求める。「Set newEv = m.AndEvent(eventA, eventB)」であれば，イベントAが起こりかつイベントBも起こっているときのイベント，という意味になる。

**＊10　OrEventメソッド**
OrEventメソッドは，「AまたはB」でイベント（DAEventオブジェクト）同士の論理和を求める。「Set newEv = m.OrEvent(eventA, eventB)」であれば，イベントAが起こるかまたはイベントBが起こったときのイベント，という意味になる。

（＝&H10000＋&H25）の[←]キーが押されたときのイベントを定義します。

```
Dim ev As DAEvent
Set ev = m.KeyDown(&H10025)
```

## 繰り返しのためのオブジェクトの定義〜CreateObjectメソッド

イベントに1回だけ反応するのではなく，繰り返し反応させる場合はどうすればいいでしょうか。そんなときは，対象となるオブジェクトを，つぎのようにCreateObjectメソッドで新しいインスタンスとして定義しておく必要があります。

```
Set roteR = CreateObject("Direct
Animation.DANumber")
```

CreateObjectメソッドの引数は，変化の対象となるDirectAnimationのオブジェクトです。DAImageオブジェクトであれば“DirectAnimation.DAImage”，DASoundオブジェクトであれば“DirectAnimation.DASound”というように，文字列であたえます。

## 繰り返しの設定〜InitメソッドとUntilメソッド

つぎに，InitメソッドとUntilメソッドを使ってオブジェクトを初期化します。

コードは多少複雑ですが，ポイントは，Untilメソッドがネストして（入れ子にして）いることと，最後に指定するオブジェクトがインスタンス化＊8したオブジェクト変数であることです。

つぎのコードは，サンプルで利用しているコードです。はじめはd1（120度）〜d2（270度）まで0.2秒で変位して，イベントevが発生したら（[←]キーが押されたら）d2（270度）〜d1（120度）まで0.2秒で変位します。

さらに，イベントevが発生したら（[←]キーが押されたら）もとにもどり，d1（120度）〜d2（270度）まで0.2秒で変位します。

```
roteR.Init m.Until(m.Interpolate
(d1, d2, 0.2), ev, _
              m.Until(m.Interpolate
(d2, d1, 0.2), ev, roteR))
```

この変動値（roteR）で，赤旗の上げ下げの回転角を制御しているのです。

## イベントは組み合わせが可能〜AndEventメソッド／OrEventメソッド

AndEventメソッド＊9やOrEventメソッド＊10を用いて，イベントと他のイベントをAndやOr条件で組み合わせることもできます。サンプルでは，キーが押された／離されたときの唇の動きを制御するために利用しています。

つぎのコードのように，[←]（&H10025）または[→]（&H10027）のどちらかが押されると発生するイベント（evDN）を定義します。

```
Set evDN = m.OrEvent(m.KeyDown
(&H10025), m.KeyDown(&H10027))
```

## イベントプロシージャを利用したキーイベントの処理

キーイベントを処理するもう一つの方法として，イベントプロシージャの利用があります。

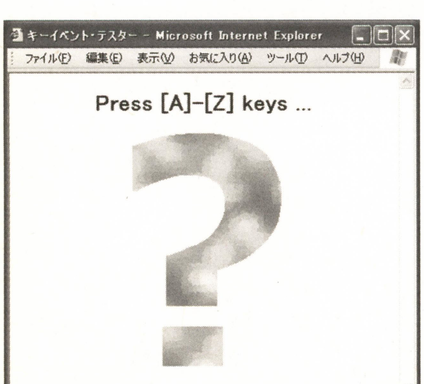
キーボードのイベントに関して，DirectAnimationコントロールのKeyDown／KeyUp／KeyPressという三つイベントプロシージャが用意されています。しかし，実際に利用できるのはKeyDownイベントプロシージャのみです。

ここでは，簡単なサンプル・プログラム『キーイベント・テスター』をもとに解説していきます。ソースコードは，**リスト2**です。

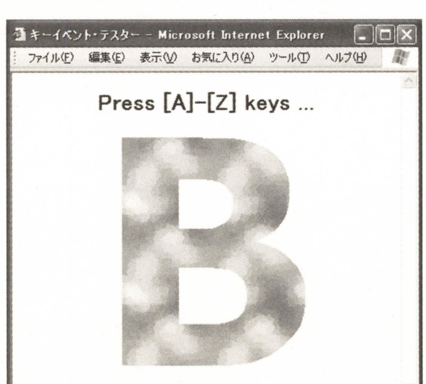
このサンプルを実行すると，**写真3**ように“？”マークが中央に表示され，文字の内部のイメージがスクロールしています。キーボードの[A]〜[Z]キーを押すと，それに反応して押したキーと同じ文字が画面上に表示されます。[B]キーを押したときは，**写真4**のように画面の文字が切り替わります。

## KeyDownイベントプロシージャに注目！

KeyDownイベントプロシージャは，DirectAnimationコントロールにフォーカスがあると

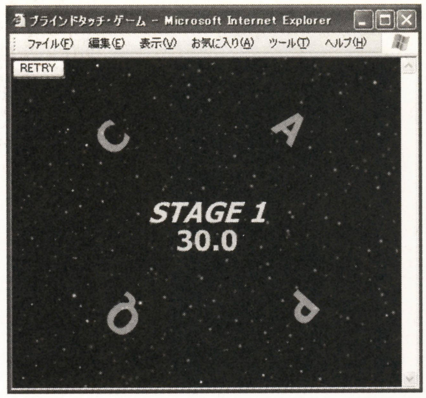


《写真3》『キーイベント・テスター』のスタート画面。[A]〜[Z]キーを押してみよう

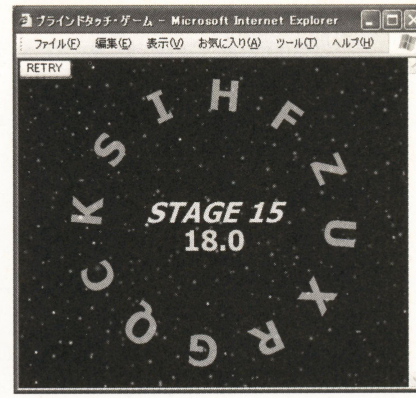


《写真4》『キーイベント・テスター』で[B]キーを押したときのようす。いろんなキーを押してみよう


《写真5》クリッピングに使用する画像ファイル



《写真6》『ブラインドタッチ・ゲーム』のスタート画面。[F]と[J]にひとさし指を置こう！



《写真7》難易度がアップすると，このように文字が増えて時間も短くなる

きに，何かキーが押されると呼びだされるプロシージャです。

DirectAnimationのKeyDownイベントプロシージャは，つぎのような書式です。

```
Private Sub DAControl_KeyDown(ByVal
KeyCode, ByVal KeyData)

End Sub
```

何のキーが押されたかは，引数のKeyCodeを調べればわかります。キーとKeyCodeの対応は，つぎのようなコード記述して，実際に動かしてみてください。

```
Private Sub DAControl_KeyDown(ByVal
KeyCode, ByVal KeyData)
   MsgBox KeyCode & " (&H" &
Hex(KeyCode) & ")"
End Sub
```

## 差し替え可能なDAStringオブジェクトを使用

キーが押されたときに，押されたキーの文字を表示するために，前述の差し替え可能なDAStringオブジェクトを利用しています。サンプルのイベントプロシージャでは，まず，つぎのようにKeyCodeからChr関数*11で文字(String型)に変換します。

```
Dim c As String
c = Chr(KeyCode)
```

あらかじめModifiableStringメソッドで定義した差し替え可能なDAStringオブジェクト(abc)に対して，SwitchToStringメソッドで文字列(c)に差し替えます。

```
abc.SwitchToString c
```

## 文字の描画はクリッピングのテクニック

描画される文字の内部は，写真5のような“BCG_047.GIF*12”ファイルをロードしてタイリング加工し，スクロール処理をしたイメージ(panel)から，文字列の軌跡で切り抜いたものです。

このようなイメージのクリッピング処理*13は，本誌2001年9月号でくわしく解説しているので，参考にしてください。

## 今月もサンプル・ゲームで遊ぼう！

『ブラインドタッチ・ゲーム』は，その名前のとおり，ブラインド・タッチの練習用ゲームです。かなり凝ったサンプル・ゲームですから，ぜひ遊んでみてください。

今回紹介したKeyDownイベントプロシージャを使って，押されたキーを判断しています。また，差し替え可能なオブジェクトして，DAImageオブジェクトとDANumberオブジェクトを利用しています。ソースコードは，リスト3です。

**●操作方法とルール**

サンプルを実行すると，写真6のような画面が表示されます。プレイヤーは，円形に表示されたアルファベットと同じキーを押してください。押されたキーは画面から消え，すべてのアルファベットを消すと，つぎのステージに進みます。

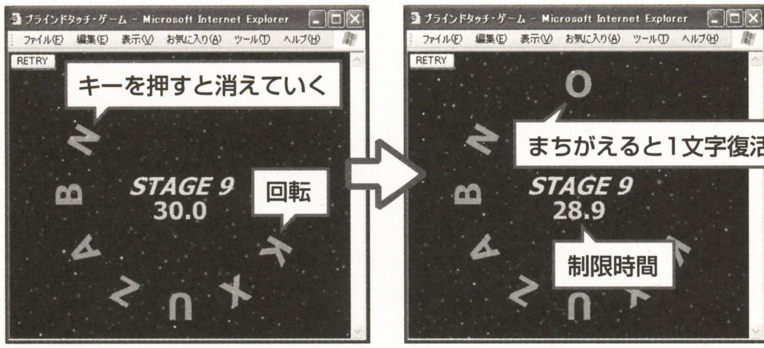
ステージが進むにつれて，写真7のように表示されるアルファベットの数が増えていきます。アルファベットが最大出題数の12個になると，つぎは制限時間がステージごとに2秒ずつ短くなっています。制限時間の最小値10秒まで進むと，今度はアルファベットの回転速度が増していきます。なお，ステージはエンドレスです。

**＊11　Chr関数**
引数に指定した数値(文字コード)に対応した文字列(String)を返すVBScriptの関数。「Chr(65)」は“A”を返す。これと反対に，文字を文字コードに変換するには Asc関数を使う。「Asc("Z")」は“90”を返す。

**＊12　BCG_047.GIF**
このGIFファイル“BCG_047.GIF”は，株式会社インプレス『Hybrid版Internetホームページ素材集 Cool＆ビジネス編』から使用しています。データの著作権は株式会社インプレスに帰属します。

**＊13　クリッピング処理**
クリッピングとは，“切り抜く”という意味。図形や文字の形でイメージを切り取ることをいう。DirectAnimationでは，DAMatteというクリッピング領域を管理するオブジェクトが用意されている。DAMatteオブジェクトは，DAPath2オブジェクトから定義される。

## 『ブラインドタッチ・ゲーム』の遊びかた




《第6図》だんだん文字が増えていき，スピードも上がってくる。ブラインドタッチ（キーを見ないで入力すること）ができない人は，このゲームでぜひ習得しよう！

まちがったキーや一度クリアしたアルファベットのキーを押してしまうと，**第6図**のようにペナルティとして1文字復活します。

制限時間内に，表示されているすべてのアルファベットを消さないと，**写真8**のようにゲーム・オーバーとなります。もう一度はじめから開始する場合は，画面左上の[RETRY]ボタンをクリックしてください。

このサンプルで使用しているイメージ・ファイルは，縦スクロールさせる背景用のJPEGファイル*14 “stars.jpg” です（**写真9**）。

### ステージの推移はDAImageオブジェクトの差し替え

このサンプルは，ステージの切り替えに「差し替え可能なDAImageオブジェクト（stgImg）」を利用しています。

DAControl_KeyDownイベントプロシージャで，表示されているアルファベットすべてが消され，クリアしたと判断すると，nextStageプロシージャをコールしてつぎのステージのイメージ（newImg）を構築します。そして，最終的に下記コードでイメージを差し替え，つぎのステージがはじまるという仕組みです。

```
stgImg.SwitchTo newImg
```

### 押したキーと同じアルファベットが消える！

出題されているアルファベットと同じキーを押すと，そのアルファベットが消えていきますね。個々のアルファベットのイメージの透明度を設定するOpacityAnimメソッドに，差し替え可能なDANumberオブジェクトを使っています。

スタート時は0.7の半透明にしておき，KeyDownイベントプロシージャで対応するイメージがじょじょに消えるように，Interpolateメソッドで作成したものに差し替えます。透明度を指定するためのDANumberオブジェクトは，アルファベットの数と同じ要素数の配列変数opcで管理しています。

いきなり消えないで，ちょっと光ってジワーンと消えるようにしてあるのがミソです。

```
opc(n).SwitchTo m.Interpolate(1, 0, 0.5)
```

いきなり消すなら，「opc(n).SwitchToNumber 0」でOKです。

### クリア判定はどうする？

どのアルファベットが消されたか／残っているかは，出題されたアルファベットと同じ要素数の配列変数finで管理しています。

ステージの開始時は，すべて要素にFalseをセットしておき，KeyDownイベントプロシージャで該当するキーが押されたらTrueをセットします。そして，isClearプロシージャで配列finを調べ，全要素がTrueになっていればステージがクリアしたと判断します。

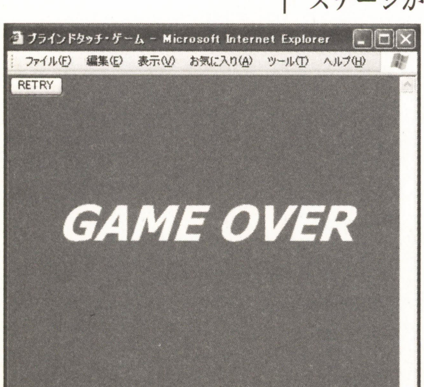
### 制限時間の表示

画面上に表示される制限時間は，変動するDANumberオブジェクトをToStringメソッド*15でDAStringオブジェクトに変換して，さらにDAImageオブジェクトとして描画しています。

はじめのアルファベットが拡大してくる2秒間は，タイマーを停止させておくため，m.DANumber(limit)をDurationメソッド*16で2秒間継続させます。そして，2秒経過すると，制限

**＊14　JPEGファイル**
JPEGは，【Joint Photographic Coding Experts Group】の略。画像圧縮アルゴリズムを制定する目的で設立された団体の略だが，JPEGは静止画像を圧縮する技術も指している。人間の目が，低周波成分については感度がよいが，高周波成分で鈍いという視覚特性を利用している。とくに，写真などのフルカラー画像を対象とした圧縮方式で，圧縮率が高いわりに画質の劣化が少ない。処理に時間がかかるのが欠点。

**＊15　ToStringメソッド**
DANumberオブジェクトをDAStringオブジェクトへ変換するメソッド。VBScriptでのCStr関数やFormatNumber関数に似ている。ToStringメソッドの引数は，小数点以下何桁まで必要なのかをあたえる。たとえば，DANumberオブジェクト型の変数numを小数点以下3桁までの文字列（DAStringオブジェクト）へ変換するには，「num.ToString(3)」と記述すればよい。




《写真8》時間が0になるとゲーム・オーバー



《写真9》『ブラインドタッチ・ゲーム』の背景用JPEGファイル

時間（limit）から0.0までlimit秒でカウント・ダウンをはじめます。

```
Set tmrStr = m.Sequence( _
m.DANumber(limit).ToString(1).Du
ration(2), _
m.Interpolate(limit, 0, limit).
ToString(1))
```

## ゲーム・オーバーの処理

ステージ・イメージ（abcImg）とゲーム・オーバー・イメージ（endImg）は，Sequenceメソッドで連結させています。

```
Set newImg = m.Sequence(abcImg.
Duration(limit + 2), endImg)
```

ステージ・イメージが制限時間（limit＋2）秒間表示された後，ゲーム・オーバー・イメージが表示されるように仕組んであります。

ゲーム・オーバー時のキーイベントの無効化は，変数alarmで管理しています。各ステージの開始時に，そのステージの終了時刻をセットします。KeyDownイベントプロシージャで現時刻がalarm以下かどうかを調べて，もし過ぎていたらイベントプロシージャをすぐに脱出させてしまいます。

このため，ゲーム・オーバー後には，どのキーを押しても反応しません。

## まとめ

今回も，前回に引き続きゲームっぽい動きをするアニメーションを作ってみました。いかがでしたでしょうか。

記載したサンプルのソースコードや画像ファイルなどは，すべてベーマガ・ホームページにアップロードしています。ぜひ，ダウンロードして動かしてみてください。

●http://www.basicmagazine.net/

**＊16　Durationメソッド**
指定期間オブジェクトを継続させるメソッド。引数は継続させる時間を，秒単位で指定する。たとえば「img.Duration(1.5)」とすると，オブジェクトimgは1.5秒間継続する。Sequenceメソッドと組み合わせて「○○を△△秒間継続し，その後◎◎へ移行する」のような動きや変化を定義する。

**リスト1～リスト3
入力&実行方法**

このリストは，IE5／5.5とテキスト・エディタ（『メモ帳』など）があれば，実行することが可能だ。IE5は，以下のアドレスから無償でダウンロードすることができる。
●http://www.microsoft.com/windows/ie_intl/ja/download/
リストをメモ帳などで入力したら，拡張子を"htm"または"html"にして保存する。そして，IE5の[ファイル]メニューより，[開く]で保存したファイルを読み込もう。すると，リストが実行されるようすを，IE5で確認することができる。

《リスト1》サンプル『レッド＆ホワイト フラッグ』"RedWhite.htm"（VBScript用）

```
<HTML>
<TITLE>レッド＆ホワイト フラッグ</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
  <CENTER><H2><B>
  <FONT NAME = "Tahoma">Press left key or right key.</FONT>
  </H2></B></CENTER>
<SCRIPT LANGUAGE = "VBScript">
'//-------------------------------------------------------
'//
'// DirectAnimation Sample                      by Masato Hirai
'//
'//    『レッド＆ホワイト フラッグ』(RedWhite.htm)
'//
'//-------------------------------------------------------
Option Explicit

'//-------------------------------------------------------
'// 初期化処理
'//-------------------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数・定数の宣言
  '//
  Const PI = 3.14159265359        '円周率
  Dim m      'As DAStatics        'DAStaticsオブジェクト
  Dim border 'As DALineStyle      '枠線スタイル
  Dim panel  'As DAImage          '塗りつぶしイメージ
  Dim ev     'As DAEvent          'DAEventオブジェクト
  Dim roteL  'As DANumber         '赤旗の上げ下げ用の変動値
  Dim roteR  'As DANumber         '白旗の上げ下げ用の変動値
  Dim numY   'As DANumber         '唇の動きを制御する変動値
  Dim pth    'As DAPath2          '唇の軌跡(半円)
  Dim handL  'As DAImage          '左腕(赤旗)のイメージ
  Dim handR  'As DAImage          '右腕(白旗)のイメージ
  Dim hand   'As DAImage          '両腕のイメージ
  Dim face   'As DAImage          '顔のイメージ
  Dim lips   'As DAImage          '唇のイメージ
  Dim eyes   'As DAImage          '目のイメージ
  Dim body   'As DAImage          '胴体のイメージ
  Dim foot   'As DAImage          '足のイメージ
  Dim back   'As DAImage          '背景イメージ
  Dim img    'As DAImage          '最終合成イメージ
  Dim tmp    'As DAImage          '作業用DAImageオブジェクト
  Dim evDN   'As DAEvent          'KeyDownイベント
  Dim evUP   'As DAEvent          'KeyUpイベント
  Dim d1     'As Double           '旗を上げた時の回転角
  Dim d2     'As Double           '旗を下げた時の回転角
  '//
  '// DirectAnimationの初期化 (ピクセル座標系)
  '//
  Set m = DAControl.PixelLibrary
  '//
  '// イベントを使用して赤旗の上げ下げを制御する変動値の定義
  '//
  d1 = PI / 6 * 4                '120度(旗を上げた時の回転角)
  d2 = PI / 6 * 7                '210度(旗を下げた時の回転角)
  Set ev = m.KeyDown(&H10025) '[←]キーを押した時のイベント
  Set roteL = CreateObject("DirectAnimation.DANumber")
  roteL.Init m.Until(m.Interpolate(d1, d2, 0.2), ev, _
          m.Until(m.Interpolate(d2, d1, 0.2), ev, roteL))
  '//
  '// イベントを使用して白旗の上げ下げを制御する変動値の定義
  '//
  Set ev = m.KeyDown(&H10027) '[→]キーを押した時のイベント
  Set roteR = CreateObject("DirectAnimation.DANumber")
  roteR.Init m.Until(m.Interpolate(d1, d2, 0.2), ev, _
          m.Until(m.Interpolate(d2, d1, 0.2), ev, roteR))
  '//
  '// 赤旗を持つ左腕のイメージの作成
  '//
  Set border = m.DefaultLineStyle.Width(2)
  Set panel = m.SolidColorImage(m.ColorHsl(0.09, 0.87, 0.8))
  Set handL = m.Rect(128, 10).Fill(border, panel)
  Set handL = handL.Transform(m.Translate2(64, 0))
  Set panel = m.SolidColorImage(m.Red)
  Set tmp = m.Rect(60, 60).Fill(border, panel)
  Set tmp = tmp.Transform(m.Translate2(128 - 30, -25))
  Set handL = m.Overlay(tmp, handL)
  Set handL = handL.Transform(m.Rotate2Anim(roteL))
  Set handL = handL.Transform(m.Translate2(-25, -35))
  '//
  '// 白旗を持つ右腕のイメージの作成
  '//
  Set panel = m.SolidColorImage(m.ColorHsl(0.09, 0.87, 0.8))
  Set handR = m.Rect(128, 10).Fill(border, panel)
  Set handR = handR.Transform(m.Translate2(64, 0))
  Set panel = m.SolidColorImage(m.White)
  Set tmp = m.Rect(60, 60).Fill(border, panel)
  Set tmp = tmp.Transform(m.Translate2(128 - 30, -25))
  Set handR = m.Overlay(tmp, handR)
  Set handR = handR.Transform(m.Rotate2Anim(roteR))
  Set handR = handR.Transform(m.Translate2(-25, -35))
  'X軸方向に-1倍することで左右反対のイメージを作成
  Set handR = handR.Transform(m.Scale2(-1, 1))
  '//
  '// 左腕と右腕のイメージの合成
  '//
  Set hand = m.Overlay(handL, handR)
  '//
  '// キー([←]か[→])を押した時に口が開く/閉じるための変動値
  '//
  Set evDN = m.OrEvent(m.KeyDown(&H10025), m.KeyDown(&H10027))
  Set evUP = m.OrEvent(m.KeyUp(&H10025), m.KeyUp(&H10027))
  Set numY = CreateObject("DirectAnimation.DANumber")
  numY.Init m.Until(m.Interpolate(32, 10, 0.2), evDN, _
          m.Until(m.Interpolate(10, 32, 0.2), evUP, numY))
  '//
  '// 変動値を用いて唇の軌跡を定義(半円の高さを変動させる)
  '//
  Set pth = m.PieRadiansAnim(m.DANumber(PI), _
                             m.DANumber(PI * 2), _
                             m.DANumber(32), _
                             numY)
  '//
  '// 唇のイメージの作成
  '//
  Set panel = m.SolidColorImage(m.Red)
  Set lips = pth.Fill(border, panel)
  Set lips = lips.Transform(m.Translate2(0, -60))
  '//
  '// 顔のイメージ(頭＋目＋唇)
  '//
  Set panel = m.SolidColorImage(m.ColorHsl(0.09, 0.87, 0.8))
  '頭のイメージ
  Set tmp = m.Oval(60, 60).Fill(border, panel)
  Set face = tmp.Transform(m.Translate2(0, -70))
  '目のイメージ(左右)
  Set panel = m.SolidColorImage(m.Black)
  Set tmp = m.Oval(10, 12).Fill(border, panel)
  Set eyes = m.Overlay(tmp.Transform(m.Translate2(-12, -78)), _
                       tmp.Transform(m.Translate2(12, -78)))
  'イメージの合成(唇＋目＋顔)
  Set face = m.OverlayArray(Array(lips, eyes, face))
  '//
  '// 胴体のイメージの作成
  '//
  Set panel = m.SolidColorImage(m.Blue)
  Set body = m.RoundRect(60, 100, 50, 50).Fill(border, panel)
  '//
  '// 足のイメージの作成
  '//
  Set panel = m.SolidColorImage(m.Black)
  Set tmp = m.Rect(20, 70).Fill(border, panel)
  Set foot = m.Overlay(tmp.Transform(m.Translate2(-15, 70)), _
                       tmp.Transform(m.Translate2(15, 70)))
  '//
  '// 背景イメージの作成(うすピンク色)
  '//
  Set back = m.SolidColorImage(m.ColorHsl(0.95, 0.5, 0.8))
  '//
  '// 全イメージの合成
  '//
  Set img = m.OverlayArray(Array(face, body, hand, foot, back))
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = img
  DAControl.Start
  Call Window_onResize        '← リサイズイベントを呼び出し
  DAControl.Focus             '← コントロールにフォーカスを移動
End Sub

'//-------------------------------------------------------
'// ウィンドウのサイズ変更時の処理
'//-------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

## 《リスト2》サンプル『キーイベント・テスター』“KeyTest.htm”（VBScript用）

```
<HTML>
<TITLE>キーイベント・テスター</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
  <CENTER><H2><B>
  <FONT NAME = "Tahoma">Press [A]-[Z] keys ...</FONT>
  </H2></B></CENTER>
<SCRIPT LANGUAGE = "VBScript">
'//------------------------------------------------------------
'//
'// DirectAnimation Sample                        by Masato Hirai
'//
'//   『キーイベント・テスター』 (KeyTest.htm)
'//
'//------------------------------------------------------------
Option Explicit
'//
'// パブリック変数の宣言
'//
Dim m      'As DAStatics        'DAStaticsオブジェクト
Dim abc    'As DAString         'DAStringオブジェクト(差し替え用)

'//------------------------------------------------------------
'// 初期化処理
'//------------------------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数の宣言
  '//
  Dim panel  'As DAImage       'クリッピング用のイメージ
  Dim fnt    'As DAFontStyle   'フォントスタイル
  Dim pth    'As DAPath2       '文字列の軌跡(DAPath2)
  Dim moji   'As DAImage       '文字イメージ
  '//
  '// DirectAnimationの初期化 (ピクセル座標系)
  '//
  Set m = DAControl.PixelLibrary
  '//
  '// 差し替え可能なDAStringオブジェクトの定義
  '//
  Set abc = m.ModifiableString("?")   '← はじめは[?]
  '//
  '// フォントの定義 (Tahoma, 72ポイント, 太字)
  '//
  Set fnt = m.Font("Tahoma", 72, m.White).Bold
  '//
  '// 文字列の軌跡の定義 & 5倍の拡大
  '//
  Set pth = m.StringPathAnim(abc, fnt)
  Set pth = pth.Transform(m.Scale2Uniform(5))
  '//
  '// クリッピング用のイメージのロード & スクロール処理
  '//
  Set panel = m.ImportImage("BCG_047.GIF").Tile  '← タイリング
  Set panel = panel.Transform(m.Translate2Rate(32, 32))
  '//
  '// 文字列の軌跡によるイメージのクリッピング
  '//
  Set moji = panel.Clip(m.FillMatte(pth))
  '//
  '// 下方に平行移動
  '//
  Set moji = moji.Transform(m.Translate2(0, 140))
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = moji
  DAControl.Start
  Call Window_onResize          '← リサイズイベントを呼び出し
  DAControl.Focus               '← コントロールにフォーカスを移動
End Sub

'//------------------------------------------------------------
'// コントロールのKeyDownイベントプロシージャ
'//------------------------------------------------------------
Private Sub DAControl_KeyDown(ByVal KeyCode, ByVal KeyData)
  Dim c      'As String          'アルファベット文字
  '//
  '// 無効なキーおよび[A]-[Z]以外の場合は脱出
  '//
  If KeyData < 0 Then Exit Sub
  If KeyCode < 65 Or KeyCode > 90 Then Exit Sub
  '//
  '// 押されたキーのKeyCodeを文字に変換
  '//
  c = Chr(KeyCode)
  '//
  '// DAStringオブジェクトの差し替え
  '//
  abc.SwitchToString c
End Sub

'//------------------------------------------------------------
'// ウィンドウのサイズ変更時の処理
'//------------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

## 《リスト3》サンプル・ゲーム『ブラインドタッチ・ゲーム』“BdTouch.htm”（VBScript用）

```
<HTML>
<TITLE>ブラインドタッチ・ゲーム</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//------------------------------------------------------------
'//
'// DirectAnimation Sample                        by Masato Hirai
'//
'//   『ブラインドタッチ・ゲーム』(BdTouch.htm)
'//
'//------------------------------------------------------------
Option Explicit
'//
'// パブリック変数・定数の宣言
'//
Const MAX_CHARS = 12              '最大出題数(<=26)
Const MIN_LIMIT = 10              '最小の制限時間(秒)
Dim m          'As DAStatics      'DAStaticsオブジェクト
Dim stgImg     'As DAImage        'ステージイメージ(差し替え可能)
Dim opc()      'As DANumber       '透明度の配列(差し替え可能)
Dim fin()      'As Boolean        'クリアフラグの配列
Dim chars      'As Long           '各ステージでの出題数
Dim ans        'As String         '出題アルファベット文字列
Dim stage      'As Long           'ステージ番号
Dim speed      'As Double         '回転速度(度)
Dim limit      'As Double         '制限時間(秒)
Dim alarm      'As Double         '終了時刻

'//------------------------------------------------------------
'// 初期化処理
'//------------------------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数の宣言
  '//
  Dim bakImg   'As DAImage       '背景イメージ
  '//
  '// 変数の初期値設定
  '//
  Randomize        '← 乱数ジェネレータの初期化
  chars = 3        '← 出題数:4問 (※開始前に+1される)
  speed = 45       '← 回転速度:毎秒45度
  limit = 30       '← 制限時間:30秒
  '//
  '// DirectAnimationの初期化 (ピクセル座標系)
  '//
  Set m = DAControl.PixelLibrary
  '//
  '// 差し替え可能なDAImageオブジェクトの定義
  '//
  Set stgImg = m.ModifiableBehavior(m.EmptyImage) '← 空イメージ
  '//
  '// 背景イメージのロード & 縦スクロール処理
  '//
  Set bakImg = m.ImportImage("stars.jpg").Tile   '← タイリング
  Set bakImg = bakImg.Transform(m.Translate2Rate(0, 16))
  '//
  '// ステージ イメージの作成
  '//
  Call nextStage  '← nextStageプロシージャの呼び出し
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = m.Overlay(stgImg, bakImg)
  DAControl.Start
  Call Window_onResize          '← リサイズイベントを呼び出し
  DAControl.Focus               '← コントロールにフォーカスを移動
End Sub

'//------------------------------------------------------------
'// コントロールのKeyDownイベントプロシージャ
'//------------------------------------------------------------
Private Sub DAControl_KeyDown(ByVal KeyCode, ByVal KeyData)
  Dim n          'As Long
  '//
  '// 無効なキーおよび[A]-[Z]以外の場合は脱出
  '//
  If KeyData < 0 Then Exit Sub
  If KeyCode < 65 Or KeyCode > 90 Then Exit Sub
  If alarm < Timer Then Exit Sub  '制限時間オーバーの場合
  '//
  '// 押されたキーのKeyCodeが出題の文字に含まれるかの審査
  '//
  For n = 0 To chars - 1      '← 出題されてた文字数分のループ
    If Chr(KeyCode) = Mid(ans, n + 1, 1) Then
      If Not fin(n) Then      '← 既にクリアしているか？
        '//
        '// アルファベットの透明度の差し替え(不透明→透明)
        '//
        opc(n).SwitchTo m.Interpolate(1, 0, 0.5)
        fin(n) = True         '← クリアフラグをON
        Exit For
      End If
    End If
  Next
  '//
  '// ペナルティの適用
  '//
  If n >= chars Then      '(出題されてた文字以外のキーの場合...)
    Call applyPenalty     '← applyPenaltyプロシージャの呼び出し
    Exit Sub
  End If
  '//
  '// クリア判定と次のステージイメージの作成
  '//
  If isClear() Then       '(すべての文字が消された場合...)
    Call nextStage        '← nextStageプロシージャの呼び出し
  End If
End Sub

'//------------------------------------------------------------
'// ステージ イメージの作成
'//------------------------------------------------------------
Sub nextStage()
  Dim abcImg   'As DAImage      'アルファベット(出題)イメージ
  Dim endImg   'As DAImage      'ゲームオーバー(終了)イメージ
  Dim newImg   'As DAImage      '最終イメージ
  Dim tmrStr   'As DAString     '制限時間(DAString)
  Dim tmp      'As DAImage      '作業用DAImageオブジェクト
  Dim num      'As DANumber     '作業用DANumberオブジェクト
  Dim fnt      'As DAFontStyle  'フォントスタイル
  Dim n        'As Double
  '//
  '// 難易度の設定(出題数→制限時間→回転数 の順に厳しくする)
  '//
  If chars < MAX_CHARS Then        '(最大出題数までは...)
    chars = chars + 1              '← 出題数を1文字ずつ追加
  Else
    If limit > MIN_LIMIT Then      '(最小制限時間までは..)
      limit = limit - 2            '← 制限時間を2秒ずつ短縮
    Else                           '(最終的に)
      speed = speed + 5            '← 回転数を毎秒5度ずつ増加
    End If
  End If
  '//
  '// ステージの設定
  '//
  ans = makePattern(chars)  '← アルファベット文字列の作成
  stage = stage + 1         '← ステージ番号のアップ
  ReDim fin(chars - 1)      '← クリアフラグ配列の再定義
  ReDim opc(chars - 1)      '← 透明度配列の再定義
  '//
  '// 差し替え可能なDANumberオブジェクトの定義
  '//
  For n = 0 To chars - 1
    Set opc(n) = m.ModifiableNumber(0.7)  '← 初期値は0.7
  Next
  '//
  '// アルファベット(出題)イメージの作成
  '//
  Set fnt = m.Font("Tahoma", 48, m.Yellow).Bold
  Set abcImg = m.EmptyImage        '← 空のイメージ
  For n = 0 To chars - 1
    'アルファベット個々のイメージの作成
    Set tmp = m.StringImage(Mid(ans, n + 1, 1), fnt)
    'アルファベットを円形に配置する処理
    Set tmp = tmp.Transform(m.Translate2(0, -120))
    Set tmp = tmp.Transform(m.Rotate2Degrees(n * 360 / chars))
    '差し替え可能な変動値を用いて透明度を設定
    Set tmp = tmp.OpacityAnim(opc(n))
    'イメージを次々に合成していく
    Set abcImg = m.Overlay(tmp, abcImg)
  Next
  '全体を回転
  Set abcImg = abcImg.Transform(m.Rotate2RateDegrees(speed))
  'ステージ開始時に0倍から1倍へ拡大
  Set num = m.Interpolate(0, 1, 2)
  Set abcImg = abcImg.Transform(m.Scale2UniformAnim(num))
  '//
  '// ステージ番号イメージの作成
  '//
  Set fnt = m.Font("Tahoma", 32, m.Cyan).Bold.Italic
  Set tmp = m.StringImage("STAGE " & stage, fnt)
  Set abcImg = m.Overlay(tmp, abcImg)    '← 出題イメージに合成
  '//
  '// 制限時間イメージの作成
  '//
  Set fnt = m.Font("Tahoma", 32, m.Green).Bold
  '動的なDAStringオブジェクトを作成
  Set tmrStr = m.Sequence( _
           m.DANumber(limit).ToString(1).Duration(2), _
           m.Interpolate(limit, 0, limit).ToString(1))
  'DAStringオブジェクトからDAImageオブジェクトへ
  Set tmp = m.StringImageAnim(tmrStr, fnt)
  Set tmp = tmp.Transform(m.Translate2(0, 30))
  Set abcImg = m.Overlay(tmp, abcImg)     '← 出題イメージに合成
  '//
  '// ゲームオーバー(終了)イメージの作成
  '//
  Set fnt = m.Font("Tahoma", 56, m.White).Bold.Italic
  Set endImg = m.StringImage("GAME OVER", fnt)
  Set endImg = m.Overlay(endImg, m.SolidColorImage(m.Red))
  '//
  '// ステージイメージとゲームオーバーイメージの連結
  '//
  Set newImg = m.Sequence(abcImg.Duration(limit + 2), endImg)
  alarm = Timer + limit + 2              '← 終了時刻をセット
  '//
  '// ステージイメージの差し替え(画面全体を切り替える)
  '//
  stgImg.SwitchTo newImg
End Sub

'//------------------------------------------------------------
'// アルファベットがランダムに並んだ文字列を返す
'//------------------------------------------------------------
Function makePattern(ByVal max) 'As String
  Dim n       'As Long
  Dim c       'As String
  For n = 1 To max                         '← 文字数分のループ
    Do
      c = Chr(Rnd * 25 + 65)               '← [A]-[Z]ランダム
      If InStr(makePattern, c) <= 0 Then   '← ダブりの審査
        makePattern = makePattern & c      '← ダブってなければOK
        Exit Do
      End If
    Loop
  Next
End Function

'//------------------------------------------------------------
'// 各ステージのクリア判定 (True/Falseで返す)
'//------------------------------------------------------------
Function isClear() 'As Boolean
  Dim n       'As Long
  isClear = False
  For n = 0 To chars - 1
    If Not fin(n) Then Exit Function       '← クリアしていない
  Next
  isClear = True                           '← クリアしている
End Function

'//------------------------------------------------------------
'// ペナルティ処理 (既に正解した中から1文字を復活させる)
'//------------------------------------------------------------
Sub applyPenalty()
  Dim n       'As Long
  For n = 0 To chars - 1
    If fin(n) Then
      fin(n) = False                       '← クリアフラグをOFF
      opc(n).SwitchToNumber 0.7            '← 透明度の差し替え
      Exit For
    End If
  Next
End Sub

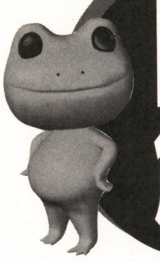
'//------------------------------------------------------------
'// ウィンドウのサイズ変更時の処理
'//------------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

'//------------------------------------------------------------
'// [RETRY]ボタン
'//------------------------------------------------------------
Sub Button1_onClick
  Window.Navigate("BdTouch.htm")       '← リフレッシュ
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
<! RETRY ボタン>
<INPUT ID="Button1" type=button value="RETRY" name=Button1>

</BODY>
</HTML>
```

プログラマーをめざす人の必見講座

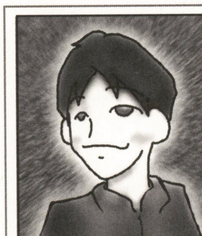


# 図解！ Visual Basic ゲーム作成術

## 第9回　キャラクター・アクション・ゲーム③追跡動作

**多くのキャラクター・アクション・ゲームでは，敵キャラクターがプレイヤー・キャラクターを追いかけるように移動します。プレイヤー・キャラクターを追いかけさせることで，ゲームの難易度を高めたり，逆にプレイヤーに敵キャラクターの移動方向を予想できるようにする効果があります。このような処理を行う場合は，“追跡動作”のアルゴリズムが必要になります。今回は，『Microsoft Visual Basic（以下VBと略）』で追跡アルゴリズムをプログラムに組み込んで，敵キャラクターにプレイヤー・キャラクターを追いかけさせてみましょう。**


TEXT&PROGRAMMING●東　幸太



BASIC歴15年以上のプログラマー。朝の起床がだんだんつらくなるこの季節，パソコンを目覚まし代わりに使うのがおもしろい。目覚まし時計は叩けばすぐに黙り込むが，パソコンは設定次第で，設定解除に必要な時間を長くできる。設定解除しているあいだに，目が覚めてくるのだ（笑）。


**Microsoft Visual Basic Programming System for Windows Ver.6.0（日本語版）**

オープン価格

※編集部の価格調査によると，下記価格が目安になります。

ラーニング版2万円強

（アカデミック・パッケージ1万円弱）

プロフェッショナル版8～9万円

（アカデミック・パッケージ1万円強）

**問い合わせ先**

マイクロソフト

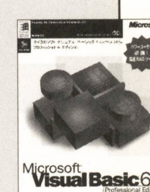
☎03-5454-2300

**ホームページ**

http://www.microsoft.com/japan/

**動作環境**

Windows 95／98／2000／NT4.0が動作するマシン





### “追跡動作”とは？

「敵キャラクターにプレイヤー・キャラクターを追いかけさせる」ということは，言い換えれば「敵キャラクターの移動方向をきめるときにプレイヤー・キャラクターを追いかけるような方向を選ぶ」となります。

仮に，ゲーム・フィールドが第1図のように何も障害物がないものであれば，プレイヤー・キャラクターとの距離が小さくなる移動方向を選べばよいのですね。しかし，一般的なキャラクター・アクション・ゲームでは，ゲーム・フィールドに壁があるためこのような簡単なアルゴリズムを使うわけにはいきません。
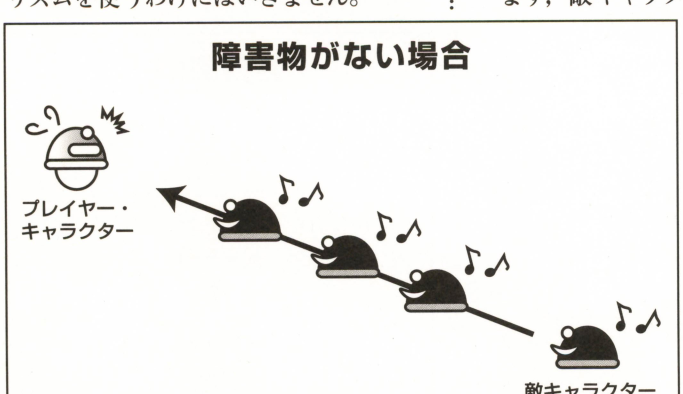
壁でさえぎられてしまうと，距離が小さくなる方向へ進む限り，壁を迂回して進めず（第2図），いつまでたってもプレイヤー・キャラクターの場所にたどり着けません。

仮に，人間が敵キャラクターを操作していたらどうするでしょうか？　壁を迂回して，第3図のように進むことでしょう。プログラムでこのような動きをさせる場合，壁を伝って進むような処理を組み込む必要があります。

### 人間の動きをアルゴリズムにする！

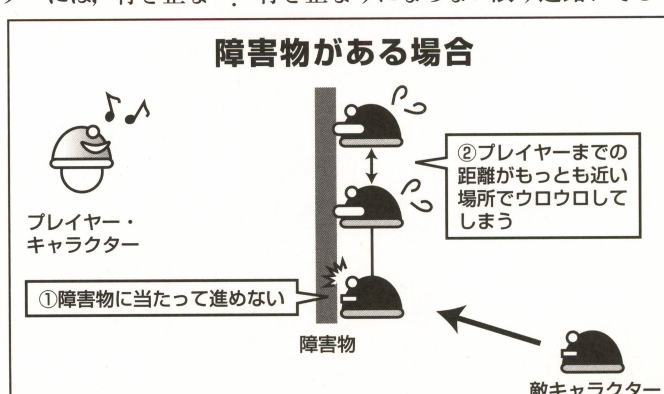
ここでは，第4図のようなアルゴリズムを考えます。

まず，敵キャラクターには，行き止まりになっている場合を除いてUターンをさせないようにします。これによって，周囲の状況が変化するまでは，きめた方向へ進むようになります。通常の通路は進める方向は二つしかありませんから，行き止まりにならない限り通路にそって



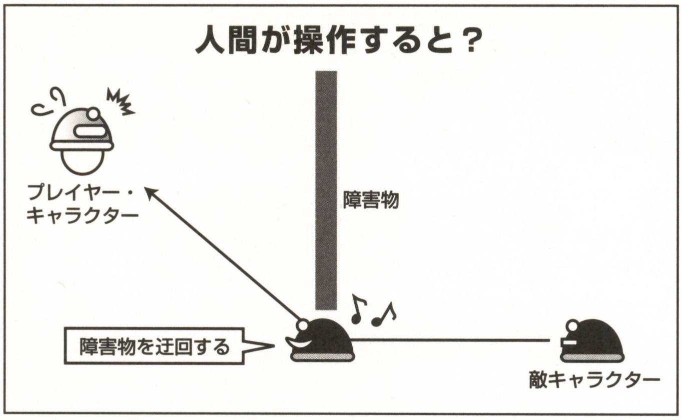


《第1図》あいだに障害物などがなければ，敵キャラクターはプレイヤー・キャラクター目がけて一直線に進んでいけばよい

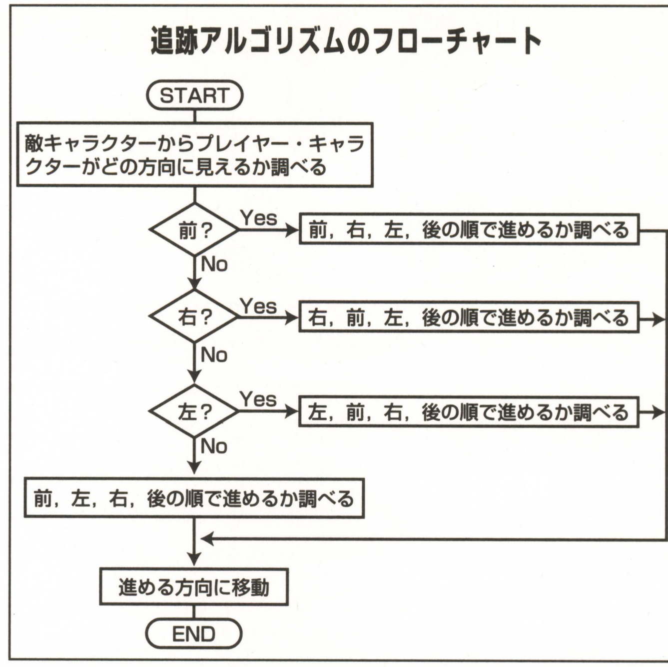




《第2図》プレイヤー・キャラクターとのあいだに障害物があると，永遠にプレイヤー・キャラクターまではたどり着かない




《第3図》「もし人間が判断するとどうなる？」ということを考えて，その判断処理をプログラムで実現させるわけだ




《第4図》敵キャラクターをUターンさせずに，プレイヤー・キャラクターのいる方向に曲がるようにするアルゴリズムがこれだ

進んでいくようになります。

つぎに，敵キャラクターから見てプレイヤー・キャラクターが右側に存在する場合は，なるべく右に曲がるようにします。逆に左側に存在する場合は，なるべく左に曲がるようにします。

これによって，**第5図**のようにプレイヤー・キャラクターが存在する方向へ敵キャラクターが方向転換するようになります。

仮に，**第6図**のような状況でプレイヤー・キャラクターが移動しなかった場合，**第4図**のようなアルゴリズムを用いれば，図のように敵キャラクターはプレイヤー・キャラクターの場所に到達できるわけです。

## 多次元テーブルの活用

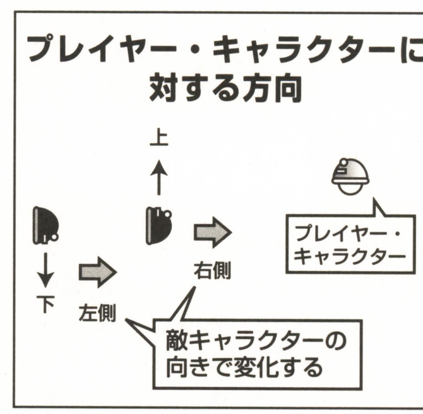
**第4図**のアルゴリズムを実際にプログラムに組み込む場合，問題になるのが「敵キャラクターから見て，右側，左側」という条件判断の部分です。

たとえば，プレイヤー・キャラクターと敵キャラクターが**第7図**のような位置関係にある場合，敵キャラクターが上を向いていればプレイヤー・キャラクターは右側になります。しかし，下を向いていると左側になってしまい，位置関係と敵キャラクターの向きによって“右側”“左側”が変わってしまいます。

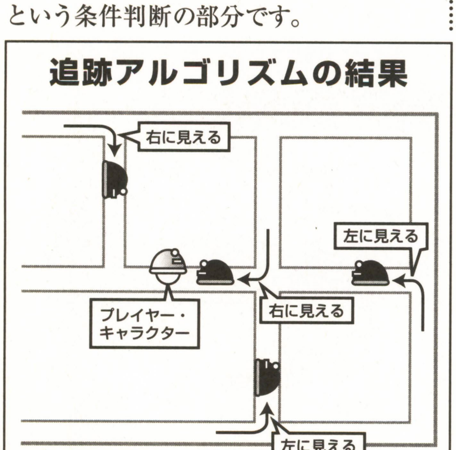
このような判断をIF文などで行おうとすると，位置関係で4とおり，敵キャラクターの向きで4とおりの条件判断になり，プログラムが複雑になってしまいます。そこで，ここで先月紹介したテーブルを活用して，プログラムを簡単にしてみます。

テーブルを活用する場合に必須なのが，条件の数値化です。敵キャラクターの向きは，先月と同様に数値に対応させます（**第8図**）。位置関係は，厳密に考えると**第9図**のようにX座標の方向について3とおり，Y座標の方向について3とおりあります。

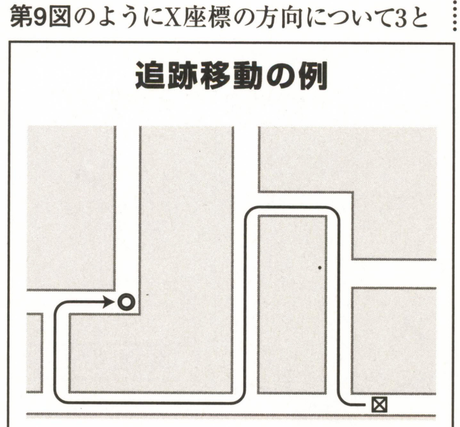
これらをすべてインデックスとするテーブルを用意し，敵キャラクターからプレイヤー・キャラクターを見た場合にど



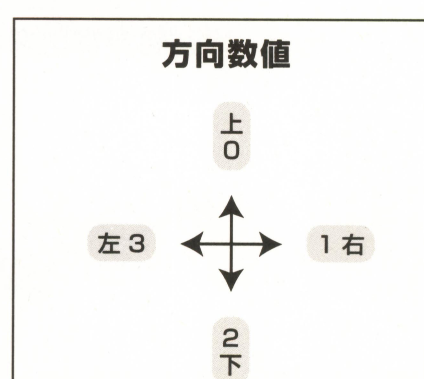

《第7図》敵キャラクターの向きによって，プレイヤー・キャラクターの見える方向が異なる。これで処理が異なることに注意しよう



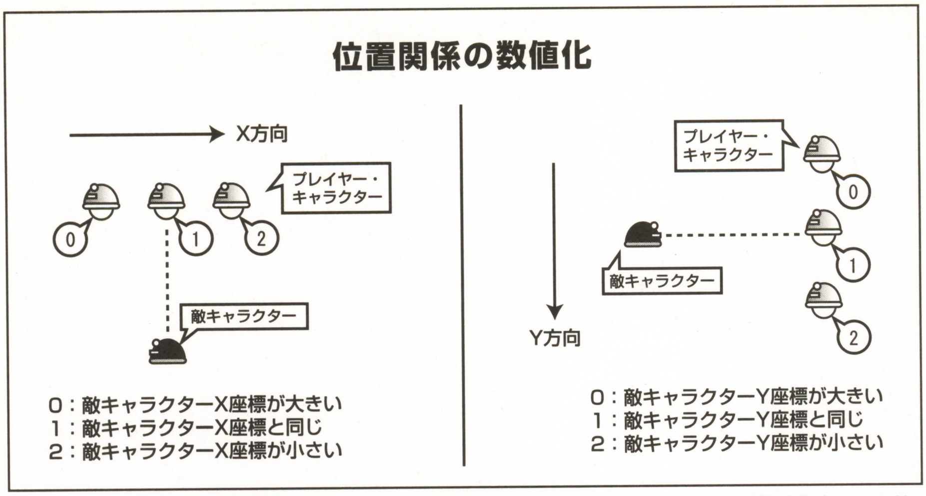

《第5図》第4図のフローチャートで処理すると，このような方向転換が実現する




《第6図》第4図の追跡アルゴリズムを使えば，このように複雑な迷路でも敵キャラクターはプレイヤー・キャラクターへ確実に到達する




《第8図》敵キャラクターの向きを，このように数値化してテーブルで処理する




《第9図》敵キャラクターとプレイヤー・キャラクターの位置関係を，X方向とY方向で数値化するわけだ

の方向に見えるかを，方向数値として取りだせるようにします。

インデックスが3系統ありますから，テーブルの次元も3次元になるため3次元の配列変数を利用します。3次元の配列変数は添え字が三つある配列変数で，三つの添え字の値が異なっていればちがう変数が対応します。DP (0, 0, 0) ／DP (0, 0, 1) ／DP (0, 1, 1) ／DP (1, 1, 1) は，それぞれ添え字の値が異なっていますから，個別の変数になるわけです。

敵キャラクターの方向数値 (0～3)，X座標方向についての位置関係数値 (0～2)，Y座標方向についての位置関係数値 (0～2) の三つを添え字とした3次元配列変数DPを用意し，敵キャラクターから見たプレイヤー・キャラクターの方向を方向数値で格納します。

たとえば，敵キャラクターがゲーム・フィールドを右 (1) に向かって移動しているとき，敵キャラクターよりX座標が大きく (2)，Y座標が小さい (0) 場所に存在するプレイヤー・キャラクターは，敵キャラクターから見て左 (3) に見えます。よって，DP (1,2, 0) には3を格納しておきます。

このようにしてテーブルDPを用意しておけば，対応するインデックスを計算するだけで，プレイヤー・キャラクターの方向を即座に求められるわけです。

なお，位置関係の数値化には，関数SGNを利用します。プレイヤー・キャラクターの座標から敵キャラクターの座標を引き，その数値をSGNにあたえると，その数値が負数ならば−1が，正数なら+1が，ゼロならば0が返ってきます。これに1を加えることで，**第9図**のような位置関係数値を計算できます。

## ゲームで敵の動きを確認してみよう！

それでは，実際にVBでプログラムされたキャラクター・アクション・ゲームを実行してみましょう。プログラムは，**リスト1**です。

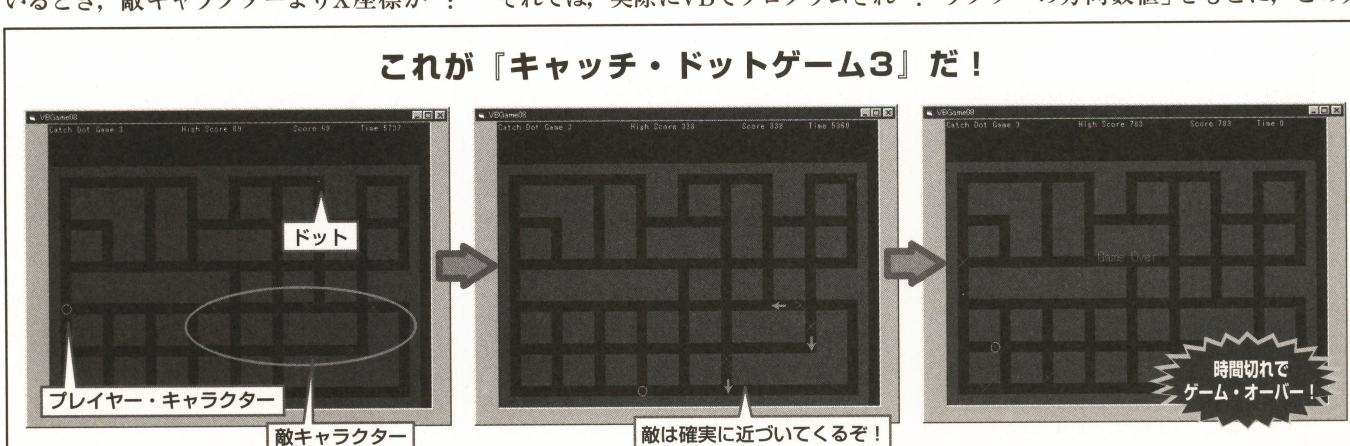
リストを入力して実行すると，**第10図**のような画面が表示されます。画面にはプレイヤーが操作するプレイヤー・キャラクターと，敵キャラクターが四つ登場します。敵キャラクターは，格子状の通路を移動してプレイヤー・キャラクターを追いかけます。移動した後に，ドットを落としていく場合があります。

プレイヤーは，プレイヤー・キャラクターをテンキーの[2][4][6][8]で上下左右に移動させ，敵キャラクターが落としたドットを制限時間内になるべくたくさん拾い集めてください。ドットを取ると，その段階での残り時間に応じた得点が加算されます。残り時間が多いほど，高得点になります。ただし，敵キャラクターと接触すると，残り時間が大幅に減ってしまいますので，注意してください。

## リストの敵追跡処理に注目！

**リスト1**では，**第4図**のアルゴリズムを利用することで，敵キャラクターがプレイヤー・キャラクターを追跡するように移動します。

敵キャラクターの移動方向を決定するときには，まず位置関係テーブルにあたえるインデックス (i0, i1, i2) を計算します。そして，このインデックスをテーブルdpにあたえ，結果として得られた「敵キャラクターから見たプレイヤー・キャラクターの方向数値」をもとに，どの方




《第10図》敵キャラクターはすべてプレイヤー・キャラクターを追跡してくるが，単純に追跡してくるため移動経路は予測しやすい。敵キャラクターの速度に注意していれば，ある程度は逃げられる。追いつめられた場合は,なるべく敵キャラクターと正面衝突し，行きちがうようにするのが有利。下手に逃げようとすると，同じ方向に移動し続けてダメージを連続して受けてしまう可能性がある

《第1表》『キャッチ・ドットゲーム3』の主な変数

| 変数名 | 内容 |
|---|---|
| hiscore | ハイスコア |
| score | プレイヤー・スコア |
| px | プレイヤー・キャラクターX座標 |
| py | プレイヤー・キャラクターY座標 |
| pm | プレイヤー・キャラクター移動カウンタ初期値 |
| pmc | プレイヤー・キャラクター移動カウンタ |
| ef() | 敵キャラクター状態フラグ |
| ex() | 敵キャラクターX座標 |
| ey() | 敵キャラクターY座標 |
| eh() | 敵キャラクターの方向 |
| em() | 敵キャラクター移動カウンタ初期値 |
| emc() | 敵キャラクター移動カウンタ |
| kg() | 仮想画面用2次元配列変数（0：空白，1：壁，2：ドット） |
| dx() | 方向→X座標移動量変換テーブル |
| dy() | 方向→Y座標移動量変換テーブル |
| eht() | 敵キャラクター移動方向検索順テーブル |
| dp() | キャラクター位置関係テーブル |
| ti | 残り時間 |
| dotc | ドットを発生させる間隔計測カウンタ |
| gamestat | ゲーム進行状況（0：ゲーム中，1：ゲーム・オーバー） |

向から移動方向を検索するかをきめます。

移動方向検索順はehtに格納されていて，添え字として方向数値をあたえるようになっています。添え字が1の場合は「右，前，左，後」の順番で，3の場合は「左，前，右，後」の順番でそれぞれ移動方向を検索します。これによって，プレイヤー・キャラクターに近づく方向に移動するようになります。

なお，ゲーム・フィールドがそれほど複雑でないため，敵キャラクターに真後ろに付かれると逃げ切れません。そこで，敵キャラクターの移動速度をやや落として振り切れるようにしてあります。

各キャラクターに移動カウンタとその初期値を用意し，処理のタイミングがきたときには，まず移動カウンタをカウントダウンさせます。そして，移動カウンタが0になったら移動処理を行い，移動カウンタを初期値にもどします。

このようにすると，移動カウンタ初期値が2であれば2回の処理タイミングで1回移動するようになり，5にすれば5回のタイミングで1回移動になります。よって，初期値の値を異なった値にすることで，移動速度を調整できるようになります。

プレイヤー・キャラクターのカウンタはpmc，初期値はpm，敵キャラクターのカウンタはemc()，初期値はem()にそれぞれ対応しています。なお，敵キャラクターの移動カウンタ初期値はキャラクターによって異なった値に設定してあり，それぞれの敵キャラクターの移動速度が異なったものになっています。

## まとめ

テーブル処理は，プログラムの簡単化，高速化にきわめて効果的です。多次元のテーブルであれば，インデックスの数値を計算するのが少々めんどうであっても，十分効果があります。

次回は，仮想画面を複数用意するテクニックを紹介します。こうすることで，複数種類のキャラクターが登場しても簡単に衝突判定を行ったり，敵キャラクターの追跡性能を高めたりすることが可能です。お楽しみに！

《リスト1》『キャッチ・ドットゲーム3』のプログラム・リスト（VB6.0用）

```
Option Explicit

Dim keystat%(9) 'テンキーの状態を記録する変数
Dim keytrig0%, keytrig1%, keytrig2%, keytrig3% 'トリガー用キ
ーの状態を記録する変数
Dim px%, py%, ph%, pf%, pmc%, pm% 'プレイヤーキャラクタ
Dim ex%(3), ey%(3), eh%(3), ef%(3), emc%(3), em%(3) '敵キャラ
クタ
Dim kg(35, 26) '仮想画面
Dim eht(3, 3) '敵キャラクタ移動方向検索テーブル
Dim dx(3), dy(3) '方向→移動量変換テーブル
Dim dp(3, 2, 2) 'キャラクタ位置関係テーブル
Dim hiscore%, score% 'ハイスコアとスコア
Dim ti% '残り時間
Dim dotc% 'ドットを発生させる間隔を数えるカウンタ
Dim gamestat% 'ゲーム進行状況

Private Sub Form_Load()
    hiscore = 0
    gameinit
End Sub

Private Sub gameinit() 'ゲーム初期化
    Dim ox%, oy%, x%, y%, xx%, yy%, i%, j%, f%, h%, c0%, c1%
    With Form1
         .Width = Screen.TwipsPerPixelX * 640 'フォームのサイ
ズを設定
        .Height = Screen.TwipsPerPixelY * 480
        .Caption = "VBGame08"
    End With
    With Picture1
         .Width = Screen.TwipsPerPixelX * 576 'ピクチャボック
スのサイズを設定
        .Left = Screen.TwipsPerPixelY * 32
        .Height = Screen.TwipsPerPixelY * 432
        .Top = 0
        .ScaleMode = 3
        .ScaleHeight = 432
        .ScaleWidth = 576
        .BackColor = RGB(0, 0, 0)
        .ForeColor = RGB(255, 255, 255)
        .FillStyle = 0
        .Font = "ＭＳ ゴシック"
        .FontSize = 10
    End With
    Picture1.Cls
    Form1.Show

    dx(0) = 0: dy(0) = -1 '移動方向→移動量のテーブル
    dx(1) = 1: dy(1) = 0
    dx(2) = 0: dy(2) = 1
    dx(3) = -1: dy(3) = 0

    dp(0, 0, 0) = 3: dp(0, 0, 1) = 3: dp(0, 0, 2) = 3 '位置関
係テーブル
    dp(0, 1, 0) = 0: dp(0, 1, 1) = 0: dp(0, 1, 2) = 2
    dp(0, 2, 0) = 1: dp(0, 2, 1) = 1: dp(0, 2, 2) = 1

    dp(1, 0, 0) = 3: dp(1, 0, 1) = 0: dp(1, 0, 2) = 1
    dp(1, 1, 0) = 3: dp(1, 1, 1) = 0: dp(1, 1, 2) = 1
    dp(1, 2, 0) = 3: dp(1, 2, 1) = 2: dp(1, 2, 2) = 1

    dp(2, 0, 0) = 1: dp(2, 0, 1) = 1: dp(2, 0, 2) = 1
    dp(2, 1, 0) = 2: dp(2, 1, 1) = 0: dp(2, 1, 2) = 0
    dp(2, 2, 0) = 3: dp(2, 2, 1) = 3: dp(2, 2, 2) = 3

    dp(3, 0, 0) = 1: dp(3, 0, 1) = 0: dp(3, 0, 2) = 3
    dp(3, 1, 0) = 1: dp(3, 1, 1) = 0: dp(3, 1, 2) = 3
    dp(3, 2, 0) = 1: dp(3, 2, 1) = 2: dp(3, 2, 2) = 3

    '敵キャラクタ移動方向検索順テーブル
    eht(0, 0) = 0: eht(0, 1) = 1: eht(0, 2) = 3: eht(0, 3) =
2
    eht(1, 0) = 1: eht(1, 1) = 0: eht(1, 2) = 3: eht(1, 3) =
2
    eht(2, 0) = 0: eht(2, 1) = 3: eht(2, 2) = 1: eht(2, 3) =
2
    eht(3, 0) = 3: eht(3, 1) = 0: eht(3, 2) = 1: eht(3, 3) =
2

    Picture1.Print "Catch Dot Game 3"

    px = 17 'プレイヤーキャラクタの初期座標
    py = 15

    pm = 7
    pmc = pm

    dotc = 100

    For i = 0 To 3
        ef(i) = 1 '存在フラグをたてる
        eh(i) = 0
        em(i) = 9 + i * 2 '敵キャラクタによって速度差を付ける
        emc(i) = em(i)
    Next

    ex(0) = 1 '敵キャラクタの初期座標
    ey(0) = 5
    ex(1) = 33
    ey(1) = 5
    ex(2) = 1
    ey(2) = 25
    ex(3) = 33
    ey(3) = 25

    '最初に壁でゲームフィールドを埋める
    For y = 4 To 26
        For x = 0 To 34
            kg(x, y) = 1 '壁に対応するコード
             Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y *
16 + 15), RGB(128, 0, 128), BF '壁表示
        Next
    Next

    '通路を作る
    For y = 5 To 25 Step 4
        For x = 1 To 33
            kg(x, y) = 0 '空白に対応するコード
             Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y *
16 + 15), RGB(0, 0, 0), BF '空白（通路）表示
        Next
    Next

    For x = 1 To 33 Step 4
        For y = 5 To 25
            kg(x, y) = 0 '空白に対応するコード
             Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y *
16 + 15), RGB(0, 0, 0), BF '空白（通路）表示
        Next
    Next

    '一部の通路を壁に戻す
    For i = 0 To 14
        f = 0
        While f = 0
            f = 1
            x = Int(Rnd(1) * 9) * 4 + 1
            y = Int(Rnd(1) * 6) * 4 + 5
            h = Int(Rnd(1) * 4)
            c0 = 0
            c1 = 0
            For j = 0 To 3
                If kg(x + dx(j), y + dy(j)) = 0 Then
                    c0 = c0 + 1
                End If
                If kg(x + dx(h), y + dy(h)) = 0 Then
                    If kg(x + dx(h) * 4 + dx(j), y + dy(h) *
4 + dy(j)) = 0 Then
                        c1 = c1 + 1
                    End If
                Else
                    c1 = 0
                End If
            Next
            If c0 < 3 Or c1 < 3 Then
                f = 0
            ElseIf kg(x + dx(h), y + dy(h)) <> 0 Then
                f = 0
            End If
        Wend
        For j = 1 To 3
            kg(x + dx(h) * j, y + dy(h) * j) = 1
            xx = x + dx(h) * j
            yy = y + dy(h) * j
```

→ つぎのページに続く

リスト1の続き

```
            Picture1.Line (xx * 16, yy * 16)-(xx * 16 + 15, yy * 16 + 15), RGB(128, 0, 128), BF '壁表示
        Next
    Next

    'プレイヤーキャラクタの出現場所の周囲は強制的に通路にする
    x = 17
    For y = 13 To 17
        kg(x, y) = 0
        Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15), RGB(0, 0, 0), BF '空白（通路）表示
    Next

    score = 0
    ti = 7000
    putscore

    gamestat = 0 'ゲームモード（0でゲーム中、1でゲームオーバー状態）

    Timer1.Interval = 10 'インターバルタイマ起動
End Sub

Private Sub Picture1_MouseUp(button%, shift%, x!, y!)
    If (button And 2) = 2 Then 'マウス右ボタンクリックで強制終了
        End
    End If
End Sub

Private Sub Picture1_KeyDown(key%, shift%)
    If key = 27 Then 'ESCキーで強制終了
        End
    End If
    If key > 96 And key < 106 Then 'テンキー設定
        keystat(key - 96) = 1
    End If
    If key = 32 Then keytrig0 = 1 'トリガキー設定
    If key = 90 Then keytrig1 = 1
    If key = 88 Then keytrig2 = 1
    If key = 13 Then keytrig3 = 1
End Sub

Private Sub Picture1_KeyUp(key%, shift%)
    If key > 96 And key < 106 Then
        keystat(key - 96) = 0
    End If
    If key = 32 Then keytrig0 = 0
    If key = 90 Then keytrig1 = 0
    If key = 88 Then keytrig2 = 0
    If key = 13 Then keytrig3 = 0
End Sub

Private Sub Timer1_Timer()
    Dim npx%, npy%, npvx%, npvy%, nph1%, nph2%, mf%, nex%, ney%
    Dim i%, f!, d%, h%, c%, k0%, k1%, nh%, i0%, i1%, i2%
    If gamestat = 0 Then '通常ゲームモード
        'プレイヤーキャラクタの処理
        pmc = pmc - 1 '移動カウンタ更新
        If pmc < 1 Then '移動するか？
            pmc = pm
            npx = px
            npy = py
            npvx = 0
            npvy = 0
            If keystat(2) = 1 Then '下キーを押しているか？
                npvy = npvy + 1
            End If
            If keystat(4) = 1 Then '左キーを押しているか？
                npvx = npvx - 1
            End If
            If keystat(6) = 1 Then '右キーを押しているか？
                npvx = npvx + 1
            End If
            If keystat(8) = 1 Then '上キーを押しているか？
                npvy = npvy - 1
            End If
            If npvx <> 0 Or npvy <> 0 Then '移動キーを押しているか？
                If npvx <> 0 And npvy <> 0 Then '移動キーを2つ押しているか？
                    If ph = 0 Or ph = 2 Then
                        nph1 = 2 - npvx '第1候補
                        nph2 = 1 + npvy '第2候補
                    Else
                        nph1 = 1 + npvy
                        nph2 = 2 - npvx
                    End If
                Else
                    If npvx <> 0 Then
                        nph1 = 2 - npvx '押されている移動キーが1つの場合は第1候補と第2候補を同じものにする
                        nph2 = 2 - npvx
                    Else
                        nph1 = 1 + npvy
                        nph2 = 1 + npvy
                    End If
                End If
                If kg(npx + dx(nph1), npy + dy(nph1)) <> 1 Then '第1候補の方向が壁で無いか？
                    mf = 1
                    ph = nph1
                ElseIf kg(npx + dx(nph2), npy + dy(nph2)) <> 1 Then '第2候補の方向が壁で無いか？
                    mf = 1
                    ph = nph2
                Else
                    mf = 0 '第1候補&第2候補とも壁で移動できない場合
                End If
                If mf = 1 Then '移動できる場合
                    npx = npx + dx(ph) '移動できる方向に移動
                    npy = npy + dy(ph)
                    putplayer px, py, 0
                    putplayer npx, npy, RGB(255, 255, 255)
                    px = npx '座標更新
                    py = npy
                    If kg(px, py) = 2 Then '移動先にドットがあるか？
                        kg(px, py) = 0 '仮想画面のドットを消す
                        putdot px, py, 0
                        scoreup 10 + Int(ti / 100)    '得点アップ
                    End If
                End If
            End If
        End If
        putplayer px, py, RGB(255, 255, 255)
        '敵キャラクタの処理
        For i = 0 To 3
            If ef(i) = 0 Then '存在していない場合
            ElseIf ef(i) = 1 Then '存在している場合
                emc(i) = emc(i) - 1 '移動カウンタ更新
                If emc(i) < 1 Then '移動するか？
                    emc(i) = em(i)
                    If px = ex(i) And py = ey(i) Then 'プレイヤーキャラクタとの衝突判定
                        pf = 1 'プレイヤーダメージフラグ
                    End If
                    i0 = eh(i) 'インデックス1：敵キャラクタ方向
                    i1 = Sgn(px - ex(i)) + 1 'インデックス2：プレイヤーキャラクタとのX座標関係
                    i2 = Sgn(py - ey(i)) + 1 'インデックス3：プレイヤーキャラクタとのY座標関係
                    c = dp(i0, i1, i2) '位置関係テーブル参照
                    d = -1
                    For h = 0 To 3 '移動方向テーブルによる方向を順番にチェック
                        nh = eh(i) + eht(c, h)
                        If nh > 3 Then nh = nh - 4 '相対的方向のための補正
                            k0 = kg(ex(i) + dx(nh), ey(i) + dy(nh))
                        If k0 = 0 Or k0 = 2 Then   '移動先が空白かドットならば移動
                            d = nh
                            h = 3
                        End If
                    Next
                    If d > -1 Then
                        eh(i) = d
                        nex = ex(i) + dx(eh(i)) '移動先の座標
                        ney = ey(i) + dy(eh(i))
                        putenemy ex(i), ey(i), RGB(0, 0, 0)
                        putenemy nex, ney, RGB(255, 0, 0)
                        If kg(ex(i), ey(i)) = 2 Then '元の場所にドットがあった場合
                            putdot ex(i), ey(i), RGB(255, 255, 0) 'ドットを描き直す
                        End If
                        If dotc < 1 Then '一定間隔でドットを発生させる
                            putdot ex(i), ey(i), RGB(255, 255, 0) 'ドットを描く
                            kg(ex(i), ey(i)) = 2 '仮想画面にドットを設定
                            dotc = 100
                        End If
                        dotc = dotc - 1 'ドット発生カウンタ更新
                        ex(i) = nex
                        ey(i) = ney
                        If px = ex(i) And py = ey(i) Then 'プレイヤーキャラクタとの衝突判定
                            pf = 1 'プレイヤーダメージフラグ
                        End If
                    End If
                End If
            End If
        Next
        If pf > 0 Then 'プレイヤーがダメージを受けている？
            ti = ti - 200 '残り時間を減らす
            pf = 0
        End If
        ti = ti - 1
        If ti < 0 Then ti = 0
        puttime
        If ti < 1 Then gamestat = 1 '時間切れでゲームオーバー
    ElseIf gamestat = 1 Then
        With Picture1
            .CurrentX = 230
            .CurrentY = 200
            .FontSize = 15
            .ForeColor = RGB(Rnd * 255, Rnd * 255, Rnd * 255)
        End With
        Picture1.Print "Game Over"
        If keytrig3 = 1 Then
            Timer1.Interval = 0 'インターバルタイマ停止
            gameinit
        End If
    End If
End Sub

Private Sub putscore()
    Picture1.Line (200, 0)-(369, 15), 0, BF
    With Picture1
        .CurrentX = 200
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "High Score " & hiscore
    Picture1.Line (370, 0)-(469, 15), 0, BF
    With Picture1
        .CurrentX = 370
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Score " & score
End Sub

Private Sub puttime()
    Picture1.Line (470, 0)-(575, 15), 0, BF
    With Picture1
        .CurrentX = 470
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Time " & ti
End Sub

Private Sub scoreup(up)
    score = score + up
    If hiscore < score Then hiscore = score
    putscore
End Sub

Private Sub putplayer(x, y, c)
    Picture1.Circle (x * 16 + 8, y * 16 + 8), 7, c
End Sub

Private Sub putdot(x, y, c)
    Picture1.Line (x * 16 + 7, y * 16 + 7)-(x * 16 + 8, y * 16 + 8), c, BF
End Sub

Private Sub putenemy(x, y, c)
    Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15), c
    Picture1.Line (x * 16 + 15, y * 16)-(x * 16, y * 16 + 15), c
End Sub
```

## リスト1入力&実行方法

**①VB起動**

まずは，VBを起動します。新しいプロジェクトとして，「標準EXE」を選択して[開く]ボタンをクリックしてください。通常は，最初に標準EXEが選択された状態になっているので，[開く]をクリックするだけです。すると，**写真①**のような画面が表示されます。

**②フォームの設定**

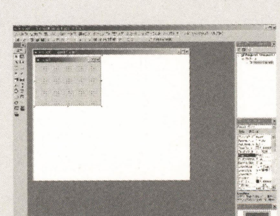
**写真②**のように，フォームを設定してください。プロパティは，プログラム中で更新されるので設定不要です（不用意に設定すると動作しなくなる場合があるので注意）。設定するのは，PictureBoxとTimerです。それぞれツールボックスのアイコンを選択した後に，フォーム上に貼り付けてください。貼り付ける操作は，フォーム上で左クリックした後，右下方向に少しドラッグしてから左ボタンを離せばOKです。位置，サイズなどは自由です（プログラム中で修正される）。

**③リストの入力**

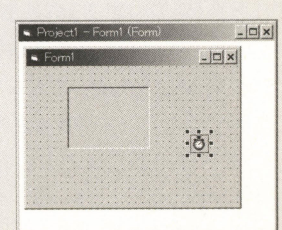
プロジェクト・ウィンドウのコード表示ボタンをクリックすると，コード・ウィンドウが表示されます。Form1のコードとして，**リスト1**を入力してください。

**④実行**

実行メニューから[開始]を選択すると，プログラムが実行されます。



《写真①》VBを起動したら「標準EXE」を選ぼう




《写真②》PictureBoxとTimerをフォームに配置する

Technical News Spot

# ネット犯罪とユーザー認証技術に注目！

テクスポ　2001年12月号

**今回は，インターネットが身近になるにつれて増えてきつつある“ネット犯罪”に注目し，将来のネットワーク社会の覇権をめぐるユーザー認証技術について紹介しましょう。**

TEXT●石川直人



ようやく引っ越しは終わったものの，未開封の段ボールが大量に積まれたまま。引っ越しでバタバタしているあいだ，ダイエット計画は頓挫しており，久しぶりに体重計に乗ったところ前月比＋3kg…。

レポート

## いたずらじゃ済まないネット犯罪

最近，つぎのようなネット犯罪が起き，ニュースとなっています。

**●他人のID利用**

先日，掲示板『2ちゃんねる』に掲載されていた他人のID，パスワードを使って，『Yahoo!オークション』にアクセスした人たち6人が逮捕されたというニュースがありました。

インターネット上にはさまざまな情報が転がっており，その中には上記のような情報も混じっています。インターネットでは，そのような情報を誰でも手に入れるチャンスがあります。

おそらく利用した人たちは，ほんのイタズラのつもりでやってしまったのでしょう。だからといって，そのような情報を悪用したら，それは犯罪です。

**●スパムメール送信**

スパムメールとは，一般に無作為に抽出されたメール・アドレス宛に出された広告メールのことを言います。スパムメールは，インターネット上では嫌われていますが，今のところスパムメールを出すこと自体は犯罪ではありません。

しかし，スパムメール送信で，先日ついに逮捕者が出ました。スパムメールを送信した罪ではなく，スパムメールの送信元として第三者の名をかたり，第三者のメールサーバに大量のエラーメールを送ることによって，業務を妨害したという罪です。

現在，まだネット関連の法整備が進んでおらず，さまざまなことがグレーゾーンの中に置かれている状況です。非常に自由度が高いインターネット上には，犯罪じみた話がいくらでも転がっています。

しかし自由であるからこそ，自らやっていいことと悪いことを判断する必要があります。みなさんも，自分自身で判断する習慣を身につけておきましょう。

ニュース

## ドットネット計画の路線変更？

このコーナーでたびたび取り上げている，マイクロソフトのドットネット計画。その要となるユーザー認証サービスが，『Passport』です。

現在のインターネットでは，さまざまなサービスごとにそれぞれのID，パスワードを使ったユーザー認証を受けなければなりません。また，オンライン・ショッピングをする場合などは，それぞれのサービスごとにクレジット・カード・ナンバーなどを入力する必要があります。

そのようなめんどうな手順を排除し，一つのユーザー認証を受けるだけで，さまざまなインターネット上のサービスを利用できるようにする。これが，Passportサービスの目的です。

マイクロソフトは，ドットネット計画の一環としてPassportサービスを推進してきました。しかし，これに対して，インターネット上のユーザー情報管理を一企業が独占することに対する批判も少なくありませんでした。

そこでマイクロソフトは，Passportサービスを他のサービスと相互に運用可能にすることを発表しました。ユーザー情報を独占することよりも，サービスの要となるユーザー認証サービスの標準を確立させることによって，ドットネット計画全体の後押しとするつもりなのでしょう。

一方，マイクロソフトの宿敵ともいえるサン・マイクロシステムズは，複数企業とともにPassportに対抗する中立的なユーザー認証技術を開発することを発表しています。

また，オープン・ソース陣営の総本山であるフリーソフトウエアファウンデーション（FSF）は，マイクロソフトのドットネット計画に対抗する技術をオープン・ソースで開発する，『DotGNU』というプロジェクトを立ち上げています。

ネットワーク中心となる次世代コンピュータ社会の覇権をめぐる戦いは，まずはユーザー認証の標準化競争という場ではじまっているようです。

**●今月の予想：インターネットが一般に広まるにつれ，ネット犯罪はこれからも増え続けていくでしょう。これは予想するまでもなく，既定の事実だと思います。Passportのような共通ユーザー認証技術は，犯罪をシステム的に防止しながらユーザーの利便性を高めるという，非常にむずかしい課題をクリアしていかなければならないのです。**

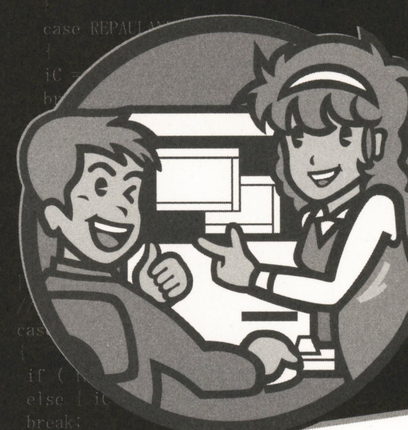
**プログラマー&クリエイターへの登竜門**

# BM特選
# 投稿プログラムコーナー

- Visual Basic
- DirectX for Visual Basic
- VBScript +DirectX
- Visual C++
- iアプリ
- PC-9801 N-88BASIC
- FX-890P
- MUSIC

本コーナーは，さまざまな言語に対応したプログラミング学習と創作をテーマに，読者からの投稿作品で成り立っています。これから，プログラミングを勉強しようというみなさんや，プロのプログラマー／クリエイターを目指そうというみなさんにとって役立つ情報が満載です。手近にあるパソコンに入力して，お楽しみください。

それぞれの記事にはプログラムを入力／実行するときの注意点やプログラムの内容，改造方法などが紹介されていますので，よく読んでください。また，“CHECKER FLAG”というコラムがあり，デバッグ博士ことDr.Dがそれぞれのプログラムに対し批評を加えています。口が悪い人なので，かなりキビシイことも言っていますが，自分でプログラムを作ったり，これから投稿しようとしている人にとっては，とても参考になるでしょう。



## プログラム投稿者へのお願い

自分の作ったプログラムを投稿しようという人はつぎのきまりを守って応募してください。

**①プログラム（作品）ファイル ②プログラムの説明，住所・氏名・年齢・職業（学校名）・電話番号（連絡のつく時間も）を書いた原稿のテキスト・ファイルをフロッピーディスクなどにセーブし（プリンタで打ち出した原稿も同封のこと），お送りください。**

プログラムの説明には，①内容 ②使いかた・遊びかた ③操作方法 ④プログラムのしくみ（変数表やフローチャートも）⑤プログラムの具体的入力方法，注意点 ⑥改造のアドバイス，作ったとき苦労した点 ⑦使用機種名 ⑧言語（バージョン，モードなども）などを書きます。

注１）プログラムは必ずチェックし，ミスを防ぐために同じものを2～3入れてもらえば安心です。

注２）プログラムの説明は，テキスト・ファイルをフロッピーディスクなどにセーブし，念のためプリンタで打ち出したものも同封することが基本ですが，手書き原稿でも受け付けます。その際，できるだけていねいに，マス目のある用紙を使用してください。

注３）ディスクのラベル，封筒の裏面にもプログラム名（ファイル名も），原稿ファイル名，住所，氏名，電話番号を必ず記入してください。

内容はオリジナルのもので，未発表のものに限ります。もし，参考にした記事や本がありましたら，書名，号数，記事名，筆者名，記事のタイトルなども明記しておいてください。

お送りいただいたディスクはお返しできませんので，あらかじめバックアップをとっておいてください。

掲載されたプログラムに対しては，規定の原稿料をお支払いし，掲載誌をお送りいたします。

**編集部**

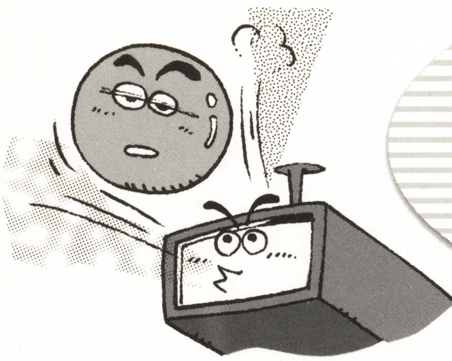
**★本誌とじ込みの愛読者カードについている応募カードも忘れずに同封してください。**

**あて先：〠141-8715 東京都品川区東五反田1-11-15　電波新聞社マイコンBASICマガジン「特選プログラム・コーナー」係**

**★ここに掲載されているプログラムは，ベーマガ・ホームページからもダウンロードできます。また，ホームページからの投稿も受け付けています。**

**http://www.basicmagazine.net/**

# Windows(Visual Basic V6.0)用



通風孔でボールを操作だ

ウィンド

# WIND

ジェイエム（18歳）

## 内容

１人テニスゲームですが，パッドは勝手に動きます。

左右にある，通風孔を使ってボールを操作して，パッドに当てるゲームです。

## 遊びかた

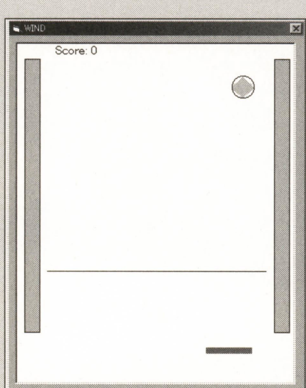
←キーを押せば，風は左に流れ始め，ボールも左に流れます。

→キーを押せば，風は右に流れ始め，ボールも右に流れます。

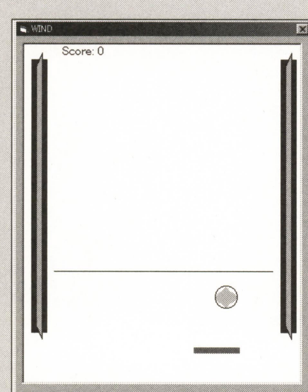
なにも押されてない状態で，通風孔が開いていたら，徐々に閉じていきます。

また，横に引かれているラインの下(ストップゾーン)にボールが入ると，パッドは止まります。

一定時間ごとに，ストップゾーンの範囲が小さくなり，だんだん難しくなります。



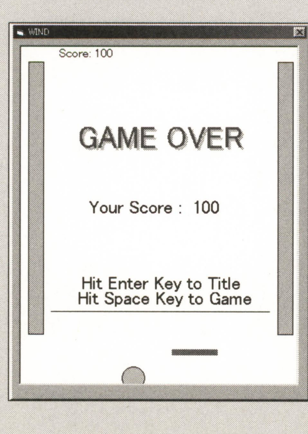

《写真1》一見、単純なゲームに見えるが…




《写真2》通風孔でボールを動かすのが特徴




《写真3》細かな調整が必要なため、意外とムズイ




《写真4》長くプレイして高得点を叩き出せ

## 入力方法

フォーム上に，TimerとPictureBoxを貼り付けてください。

つぎに，PictureBoxのオブジェクト名を，Fieldにしてからコピーして，フォームに貼り付けて，Field(0)とField(1)になるようにしてください。

プロパティは,リストで行っているので,変更する必要はありませんが,好みで,Form1のプロパティを,「BorderStyle＝3」,「StartUpPosition＝2」にしてください。

なお，改造のアドバイスとしては，音をつけたりすることです。

## CHECKER FLAG

**影**：通風孔から風を出してボールを操作するってのがおもしろいなぁ。ボールの動きも，いかにも風の影響をうけて動いているって感じだ。

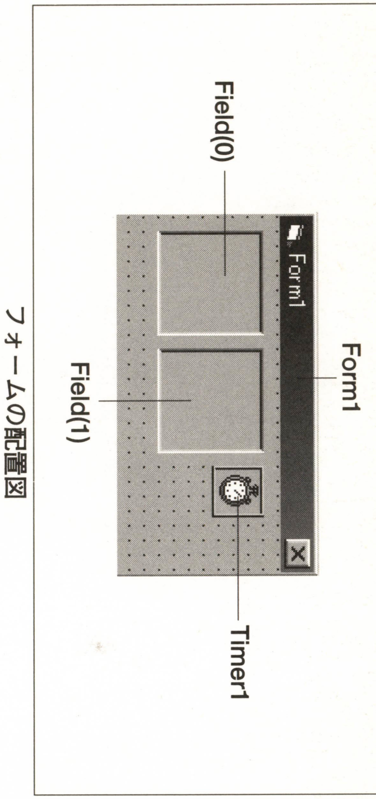
**つぐ美**：その通風孔の動きにも注目です。左右の違いによって，ちゃんと描画の区別をしているんですよね。

**編**：こういったチョットしたアイデアで，オリジナリティのあるゲームは作れます。他の投稿者の良い見本となる作品です。

**Dr.D**：影クンが触れておるボールの動きは，通風孔の開き具合によって風力(Wdで示される)を設定し,その値を利用してボールの動きを処理していることから，風の影響を受けている感じが再現されておる，という形じゃな。リスト全体もシンプルな記述ながら三角関数を巧みに利用しておることで，ボールや通風孔などを自然な動きで再現しておる。ぜひとも参考にしてほしいリストじゃな。



《OVER FLOW》月日のたつのは早いもので，2001年度も12月号となりました。この1年を振り返ってみると，ｉアプリに見られるような新たなプラットフォームでの投稿も増えました。来年度はまたどんなスタイルの投稿プログラムが誌面を賑わせるか楽しみです。




フォームの配置図

《第１表》プロパティ表

| | | | |
|---|---|---|---|
| **フォームの設定** | | Left | 120 |
| Name | Form1 | ScaleHeight | 795 |
| BorderStyle | 3 '固定ダイアログ | ScaleWidth | 795 |
| Caption | Form1 | TabIndex | 0 |
| Height | 1665 | Top | 120 |
| Left | 0 | Width | 855 |
| LinkTopic | Form1 | Name | Field(1) |
| MaxButton | 0 'False | Height | 855 |
| MinButton | 0 'False | Index | 1 |
| ScaleHeight | 1290 | Left | 1080 |
| ScaleWidth | 2910 | ScaleHeight | 795 |
| ShowInTaskbar | 0 'False | ScaleWidth | 795 |
| StartUpPosition | 2 '画面の中央 | TabIndex | 1 |
| Top | 0 | Top | 120 |
| Width | 3000 | Width | 855 |
| **ピクチャボックスの設定** | | **タイマーの設定** | |
| Nmae | Field(0) | Name | Timer1 |
| Height | 855 | Left | 2040 |
| Index | 0 | Top | 120 |

## 『WIND』のプログラム・リスト

```
Private Declare Sub Polygon Lib "gdi32" (ByVal hdc As Long, lpPoint As Any, ByVal nPoint As
Long)
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As Long, ByVal y As
Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long,
ByVal ySrc As Long, ByVal dwRop As Long) As Long
Private Declare Function GetKeyState Lib "user32" (ByVal nVirtKey As Long) As Integer

Dim Xmax, Ymax                          As Integer
Dim Score, Time, StopZone, Condition    As Integer
Dim Wx(1), Wy(1), Wr, Wrv, Wd           As Integer
Dim Px, Py, Pw, Ph, Pv, Pc              As Integer
Dim Bx, By, Bs, Bc, Brv                 As Integer
Dim PI, Br                              As Double
Dim Poly(1, 3)                          As Long

'==========================================================================================
'   < Form_Load >
'------------------------------------------------------------------------------------------
'  フォームのロード
'==========================================================================================
Private Sub Form_Load()
    With Form1
        .Width = Screen.TwipsPerPixelX * 395
        .Height = Screen.TwipsPerPixelY * 500
        .Caption = "WIND"
    End With

    Dim i As Integer

    For i = 0 To 1
        With Field(i)
            .Width = Screen.TwipsPerPixelX * 370
            .Height = Screen.TwipsPerPixelY * 450
            .Top = 120
            .Left = 120
            .BackColor = RGB(255, 255, 255)
            .FillStyle = 0
            .ScaleMode = 3

            If i = 1 Then
                .AutoRedraw = True
                .Visible = False
            End If
        End With
    Next

    GameInit
End Sub

'==========================================================================================
'   < PutWind >
'------------------------------------------------------------------------------------------
'  通風孔の描画
'==========================================================================================
Private Sub PutWind()
    Dim k As Integer
    Dim cr, sr As Double

    cr = Cos(Wr * PI / 180) * 10
    sr = Sin(Wr * PI / 180) * 10

    Field(1).ForeColor = 0
    Field(1).FillColor = RGB(200, 200, 200)

    Poly(1, 0) = Wy(0) + sr
    Poly(1, 1) = Wy(0) - sr
    Poly(1, 2) = Wy(1) + sr
    Poly(1, 3) = Wy(1) - sr

    For k = 0 To 1
        Field(1).Line (Wx(k) - 10, Wy(0))-(Wx(k) + 10, Wy(1)), 0, BF

        Poly(0, 0) = Wx(k) - cr
        Poly(0, 1) = Wx(k) + cr
        Poly(0, 2) = Wx(k) + cr
        Poly(0, 3) = Wx(k) - cr

        Polygon Field(1).hdc, Poly(0, 0), 4
    Next
End Sub

'==========================================================================================
'   < PutPad >
'------------------------------------------------------------------------------------------
'  パッドの描画
'==========================================================================================
Private Sub PutPad()
    Field(1).Line (Px - Pw, Py)-(Px + Pw, Py + Ph), RGB(255, 0, 0), BF
End Sub

'==========================================================================================
'   < PutBall >
'------------------------------------------------------------------------------------------
'  ボールの描画
'==========================================================================================
```

★ついに！金欠でハガキ＆切手が買えなくなりました。うぐっ（泣）！（東京都練馬区・**ルストフレム14歳**）……【**編**：だから，メールでの投稿なんですね。金欠病の人でも安心して投稿できるベーマガって，なんてすばらしいんでしょう。そう思いません？】

```
Private Sub PutBall()
    Field(1).FillColor = RGB(0, 255, 0)
    Field(1).Circle (Bx, By), Bs
End Sub


'==========================================================================
'   < PutText>
'--------------------------------------------------------------------------
'  文字の描画
'==========================================================================
Private Sub PutText(x, y, s, c As Long, tx As String)
    With Field(1)
        .CurrentX = x
        .CurrentY = y
        .FontSize = s
        .ForeColor = c
    End With

    Field(1).Print tx
End Sub

'==========================================================================
'   < Xref >
'--------------------------------------------------------------------------
'  X方向の反射処理
'==========================================================================
Private Function Xref(r As Double)
    r = alr(r)
    r = -r + PI
    Xref = r
End Function


'==========================================================================
'   < Yref >
'--------------------------------------------------------------------------
'  Y方向の反射処理
'==========================================================================
Private Function Yref(r As Double)
    r = alr(r)
    r = -r
    Yref = r
End Function


'==========================================================================
'   < alr >
'--------------------------------------------------------------------------
'  ラジアン角度の修正
'==========================================================================
Private Function alr(r As Double)
    While r >= PI Or r <= -PI
        If r < -PI Then r = r + PI * 2
        If r > PI Then r = r - PI * 2
    Wend
    alr = r
End Function


'==========================================================================
'   < Timer1_Timer >
'--------------------------------------------------------------------------
'  タイマーイベント (ループ)
'==========================================================================
Private Sub Timer1_Timer()
    Select Case Condition
        Case 0: GameTitle
        Case 1: GameMain
        Case 2: GameEnd
    End Select
```

```
    BitBlt Field(0).hdc, 0, 0, Xmax, Ymax, Field(1).hdc, 0, 0, vbSrcCopy
End Sub


'==========================================================================
'   < GameInit >
'--------------------------------------------------------------------------
'  ゲームの初期化
'==========================================================================
Private Sub GameInit()
    PI = 3.1415926538979

    '画面のサイズ
    Xmax = 370
    Ymax = 450

    'スコア
    Score = 0
    Time = 500
    StopZone = 150

    '通風孔のX座標
    Wx(0) = 20
    Wx(1) = Xmax - 20

    '通風孔のY座標
    Wy(0) = 20
    Wy(1) = Ymax - 70

    Wr = 180            '角度
    Wd = 0              '風力

    Px = Xmax / 2       'X座標
    Py = Ymax - 50      'Y座標
    Pv = 0              '速度
    Pw = 30             '幅
    Ph = 7              '高さ
    Pc = 0              'カウンタ

    Bx = Xmax / 2       'X座標
    By = Ymax / 2       'Y座標
    Bc = 0              'ボールとパッドの衝突判定のカウンタ
    Br = -60 * PI / 180 '角度
    Brv = 8             '速度
    Bs = 15             'サイズ

    Timer1.Interval = 1
End Sub


'==========================================================================
'   < GameTitle >
'--------------------------------------------------------------------------
'  ゲームタイトル
'==========================================================================
Private Sub GameTitle()
    Field(1).Cls

    PutText 50, 50, 100, RGB(200, 200, 200), "WIND"
    PutText 40, 40, 100, RGB(0, 200, 0), "WIND"

    PutText 80, 300, 18, 0, "Please Hit Space Key"

    If (GetKeyState(32) And &H80) <> 0 Then
        GameInit
        Condition = 1
    End If
End Sub


'==========================================================================
'   < GameMain >
```



★（10月号97ページより）そのとおり！破壊しない影さんは影さんじゃない（泣）！（東京都足立区・**オカッチ20歳**）……**【影：**では名前を変えることにいたします。（しばらく思考）影よりいい名前が浮かびません。みなさん，考えてくれませんか？よろしく！**】**

```
'-------------------------------------------------------------------------------------------
'   ゲームの中心
'===========================================================================================
Private Sub GameMain()
    Randomize Timer

    Field(1).Cls

    'キー入力チェック
    If (GetKeyState(37) And &H80) <> 0 Then
        Wrv = -1    '『←』キー
    ElseIf (GetKeyState(39) And &H80) <> 0 Then
        Wrv = 1     '『→』キー
    Else
        If Wr < 180 Then
            Wrv = 1
        ElseIf Wr > 180 Then
            Wrv = -1
        Else
            Wrv = 0
        End If
    End If

    '通風孔の角度と、風力の増減
    Wr = Wr + Wrv * 5
    Wd = Wd + Wrv

    '通風孔の角度と、風力の制御
    If Wr < 100 Then Wr = 100: Wd = -16
    If Wr > 260 Then Wr = 260: Wd = 16

    '通風孔の表示
    PutWind


    Time = Time - 1
    If Time = 0 And StopZone > 0 Then
        StopZone = StopZone - 25
        Time = 500
    End If

    'パッドの動き
    If Pc = 0 Then
        If Time > 0 And By > Ymax - StopZone Then
            Pv = 0
            GoTo jump
        End If

        Dim r As Integer
        r = Rnd(1) * 3

        If r = 1 Then
            Pv = 1
            Pc = Rnd(1) * 10
        ElseIf r = 2 Then
            Pv = -1
            Pc = Rnd(1) * 10
        End If
    Else
        Pc = Pc - 1
    End If

jump:
    If StopZone > 0 Then
        Field(1).Line (40, Ymax - StopZone)-Step(Xmax - 80, 0)
    End If

    Px = Px + Pv * 5
    If Px - Pw < 40 Then Px = Pw + 40
    If Px + Pw > Xmax - 40 Then Px = Xmax - 40 - Pw

    'パッドの表示
    PutPad

    'ボールの移動
    Bx = Bx + Cos(Br) * Brv + (Wd / 2)
    By = By + Sin(Br) * Brv

    'ボールと左壁の衝突
    If Bx - Bs < 40 Then
        Bx = 40 + Bs
        Br = Xref(Br)
    End If

    'ボールと右壁の衝突
    If Bx + Bs > Xmax - 40 Then
        Bx = Xmax - 40 - Bs
        Br = Xref(Br)
    End If

    'ボールと天井の衝突
    If By - Bs < 0 Then
        By = Bs
        Br = Yref(Br)
    End If

    'ボールとパッドの衝突
    If Bc = 0 Then
        If Abs(Px - Bx) < Pw + Bs And Abs(Py - By) < Ph + Bs Then
            '角度の変更
            Br = Yref(Br)
            Bc = 50

            Beep
            Score = Score + 10
        End If
    Else
        Bc = Bc - 1
    End If

    'ゲームオーバー
    If By + Bs > Ymax Then Condition = 2

    'ボールの表示
    PutBall

    'スコアの表示
    PutText 50, 0, 12, 0, "Score:" & Str(Score)
End Sub


'===========================================================================
'   < GameEnd >
'---------------------------------------------------------------------------
'   ゲームの終了
'===========================================================================
Private Sub GameEnd()
    PutText 78, 103, 30, RGB(200, 200, 200), "GAME OVER"
    PutText 75, 100, 30, RGB(200, 0, 0), "GAME OVER"
    PutText 90, 200, 18, 0, "Your Score : " & Str(Score)
    PutText 80, 300, 18, 0, "Hit Enter Key to Title"
    PutText 75, 320, 18, 0, "Hit Space Key to Game"

    If (GetKeyState(32) And &H80) <> 0 Then
        GameInit
        Condition = 1
    End If

    If (GetKeyState(13) And &H80) <> 0 Then
        GameInit
        Condition = 0
    End If
End Sub
```

★FOMAが首都圏で始まったのはいいが，東北地方はいつからサービス開始になるんだろう…すごく楽しみにしているのに…(;_;)。（福島県郡山市・**近内沖尋18歳**）……**【編：来年には全国規模に拡大する計画のようですが，具体的には？？？】**

# Windows(Visual Basic V6.0)用



マウスボタンの連打で千切りを極めろ!!

# 目指せ!!千切りマスター

(めざせ せんぎり)

高橋　好人（18歳）

## 内容

マウスボタンを連打して千切りマスターになろうという夢のようなゲームです（?）。

マウスボタンを，ずだだだ…って連打すると，「まな板の上で千切りしている」気になるでしょう･･･。

## 使いかた

実行して，メニューの「対象物」から千切りしたいものを選択します。選択したらメニューの「ゲーム一開始」を押して実行です。メニューを押すのが面倒なら，F2キーでゲームが始められます。

一応ハイスコアが残ります。ハイスコアは，メニューの「ゲーム一ハイスコア」で，初期化したり，表示することが可能です。

## 遊びかた

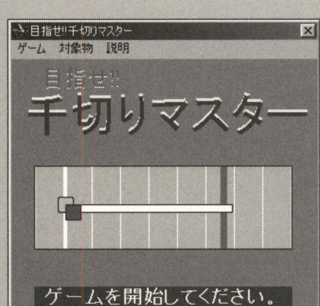
このゲームは，選択した対象物を千切りするゲームです。

ゲームを開始したらカウントが終わり次第，左の太い白線から，赤のゴールラインまで（左から右へ）マウスのボタンを連打して切っていきます。

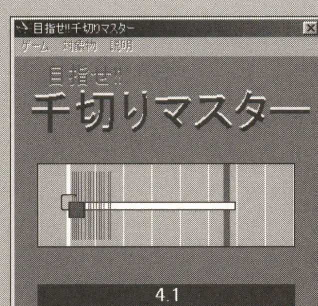
ただし，制限時間があり，10秒です。まぁ，制限時間後10秒までは，引き続きゲームプレイを延長できますが…。

対象物には，外見・長さ・硬さの３要素があり，ゲームでは，長さ・硬さが使われています。

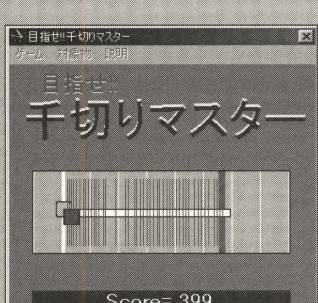
硬さというのは，千切りにした対象物の幅がその硬さに満たないとき，潰れて千切りでなくなるとしているものです。このように対象物が潰れると減点されてしまいます。大根のように硬いものは，硬さをほぼ無視できるのですが，ピーマンのようにやわらかすぎるものは，すぐに潰れてしまうのです。



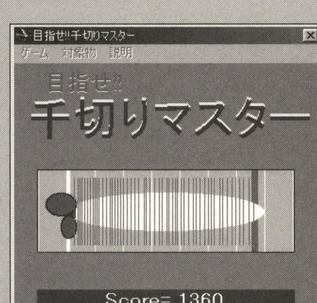

《写真１》千切りをマウス操作で体験だ




《写真２》リズミカルに切れ。気分はミスター○っ子？




《写真３》様々な要素によって得点が決まる




《写真４》千切りの対象物も複数用意

## プログラムの入力法

まずはじめに，フォームにTitleLabelと

**CHECKER FLAG**

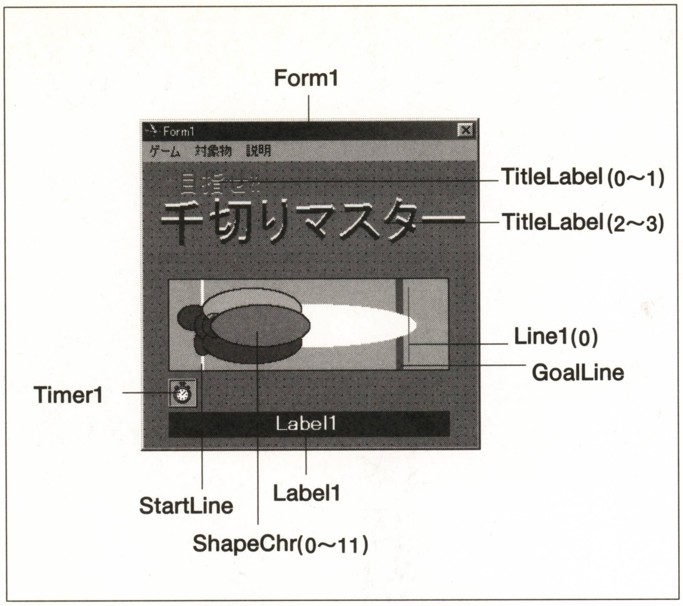
**つぐ美**：ゲーム内容は単純ですが，千切りをマウスで再現するというのがいいわね。対象物の硬さなどがあるため，得点部分などでチョットわかりにくい面はありますが，プレイするとハマる作品です。

**Dr.D**：基本操作となるマウス処理は，Main_MouseDownで行われる。内容は座標の保存やライン表示くらいなので，単純な形でゲームの操作部分が実現されておるのがわかる。あとの主な処理は，得点計算くらいなので，リストの容量はそんな大きくはないのじゃが，シェイプが多いフォームの作成が面倒そうじゃな。

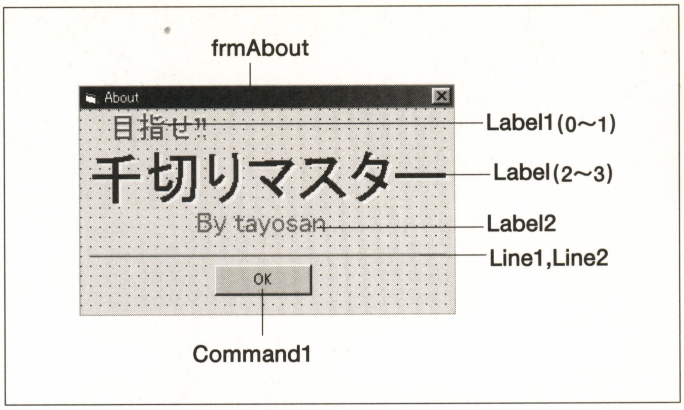
**編**：確かにそうですね。VBプログラム投稿者は，できる限りフォームのコンパクト化にも気を使ってほしいものです。



★OFでは，37歳が最高と思いきや，40歳の方もおられたので，一安心しました。ベーマガ読み始めたときは19歳だったのですが！！いつの間にやら。（島根県松江市・NSR250R＋横浜ナンバー37歳）……【影：長年愛読していただきありがとうございます。】




《第1図》フォームの配置図




《第2図》フォームの配置図




使用するカーソルの画像

いうラベルでタイトルを作ります。

つぎにMainという名のPictureを配置し，ShapeChrというシェイプを貼り付けていきます。

つぎに，Label1を配置して，メニューを入力します。frmAboutは，コマンドボタンを一つ置いてくれれば，後は自由に作ってよいでしょう。

それと，同じフォルダ内に，包丁1.cur(ピクチャ上の通常のカーソル)，包丁2.cur(ピクチャ上で切ったときのカーソル)，GameStart.WAV(ゲーム開始時のサウンド)，don.WAV(切ったときのサウンド)，GameEnd.WAV(ゲーム終了で，ハイスコアのときのサウンド)，gyu.WAV(ゲーム終了でないときのサウンド)，Kuru.WAV(対象物を変更したときのサウンド)，pp.WAV(ハイスコアを表示したときのサウンド)，bu.wav(制限時間を超過したときのサウンド)を各自で用意してください。しかし，最後から2〜3個のWAVファイルは必要ない気がするので，面倒なら用意する必要はありません。

## プログラムの仕組み

Main_MouseDownでカットした位置を記憶して，MouseMoveでゴールを超えたかを判断しています。

対象物はShapeChrというシェイプオブジェクト群で構成されていて，ObjectDataという変数に，始めと終わりのシェイプのインデックスを格納しています。

《第1表》メニューの設定

| ｷｬﾌﾟｼｮﾝ(P) | 名前(M) | ｲﾝﾃﾞｯｸｽ(X) | ﾍﾙﾌﾟｺﾝﾃｷｽﾄ番号(H) | ｼｮｰﾄｶｯﾄ(S) | ﾈｺﾞｼｴｰﾄ位置(O) | ﾁｪｯｸ(C) | 有効(E) | 表示(V) | ｳｲﾝﾄﾞｳﾘｽﾄ(W) |
|---|---|---|---|---|---|---|---|---|---|
| ゲーム | mnuGame | | 0 | (なし) | 0ーなし | | ○ | ○ | |
| …開始 | mnuGameStart | | 0 | F2 | 0ーなし | | ○ | ○ | |
| …- | mnuBar1 | | 0 | (なし) | 0ーなし | | ○ | ○ | |
| …ハイスコア | mnuHiscore | | 0 | (なし) | 0ーなし | | ○ | ○ | |
| ……ハイスコア表示 | mnuHiscoreShow | | 0 | F5 | 0ーなし | | ○ | ○ | |
| ……- | mnuBar2 | | 0 | (なし) | 0ーなし | | ○ | ○ | |
| ……ハイスコア初期化 | mnuHiscoreDelete | | 0 | (なし) | 0ーなし | | ○ | ○ | |
| …- | mnuBar3 | | 0 | (なし) | 0ーなし | | ○ | ○ | |
| …終了 | mnuEnd | | 0 | (なし) | 0ーなし | | ○ | ○ | |
| 対象物 | mnuObject | | 0 | (なし) | 0ーなし | | ○ | ○ | |
| …長葱 | mnuObjectSelect | 0 | 0 | (なし) | 0ーなし | ○ | ○ | ○ | |
| …大根 | mnuObjectSelect | 1 | 0 | (なし) | 0ーなし | | ○ | ○ | |
| …ピーマン | mnuObjectSelect | 2 | 0 | (なし) | 0ーなし | | ○ | ○ | |
| 説明 | mnuHelp | | 0 | (なし) | 0ーなし | | ○ | ○ | |
| …その他 | mnuHelpOther | | 0 | (なし) | 0ーなし | | ○ | ○ | |

## 得点について

このゲームは，結構得点の計算に力を入れました。このゲームでは減点と得点があるので，それを以下に書きます。

減点の要素は，

・硬さより細く切ったとき

・10秒を超えたとき

・一番幅が狭いのと，一番幅が広いのとの差

・幅が30ドット以上のときです。

得点の要素は，

・30ドット以下で，硬さより幅が広いとき(硬さの値にできるだけ近ければ高得点)

・10秒以下（できるだけ早ければ高得点）です。

なんとなく，減点が多いようですが，大抵得点となるほうが多いです。リストからはSub ScoreShow( )内を見てください。

## コツ

このゲームで高得点を出したい人は，読んでください。

すべての対象物の「硬さ」を頭に入れることが一番のコツです。対象物によって高得点を出す方法が異なります。

硬さは，数字が小さいほど硬いので間違えないでください。

・長葱（ながねぎ），硬さ4。一番初めに作ったもので，すべてのバランス(速さ，正確さ，切った数など)が必要で，かなり得点の上下が激しいです。コツは，やっぱりなれることです。このモードは，経験が一番ってところです。

・大根，硬さ2。このモードが最も簡単で沢山の点数が入ります。このモードは速さより，正確さと切った数が重要です。コツは10秒超えても気にせず(15秒は超えるな)，ひたすら細かく切っていくことです。

・ピーマン，硬さ7。このピーマンは包丁で切りにくい代表であるので，とても潰れやすいのです。速さと正確さが絶対条件です。よく，大根や長葱モードのように細く切る人がいますが，大抵そういう人はマイナスポイントになります。背景の白線のおよそ3分の1ぐらいのところをテンポよく，すばやく切ることがコツです。

★10月号62ページより。Kylixはいつから連載スタートするんですか？楽しみなんですけど…。（埼玉県さいたま市・**タマさんとともに期待のオタマ27歳**）……【**Dr.D**：Linux用開発ツールじゃな。C言語とともに，なんとか読者の要望にこたえようと，検討中じゃ。】

## 《第１表》Form1のプロパティ表

| フォームの設定 | |
|---|---|
| Name | Form1 |
| Appearance | 0 'フラット |
| BackColor | &H00FF8080& |
| BorderStyle | 1 '固定(実線) |
| Caption | Form1 |
| Height | 4785 |
| Icon | (アイコン) |
| Left | 105 |
| LinkTopic | Form1 |
| MaxButton | 0 'False |
| MinButton | 0 'False |
| ScaleHeight | '275 |
| ScaleMode | 3 'ピクセル |
| ScaleWidth | 324 |
| StartUpPosition | 2 '画面の中央 |
| Top | -180 |
| Width | 4950 |

| ラベルの設定 | |
|---|---|
| Name | Label1 |
| Alignment | 2 '中央揃え |
| BackColor | &H00454545& |
| Caption | Label1 |
| Font | ＭＳ Ｐゴシック |
| …Size | 16 |
| ForeColor | &H00FFFFFF& |
| Height | 25 |
| Left | 24 |
| TabIndex | 4 |
| Top | 240 |
| Width | 273 |
| Name | TitleLabel(0) |
| BackStyle | 0 '透明 |
| Caption | 目指せ!! |
| Font | ＭＳ Ｐゴシック |
| …Size | 18 |
| ForeColor | &H0000FFFF& |
| Height | 33 |
| Index | 0 |
| Left | 32 |
| TabIndex | 0 |
| Top | 8 |
| Width | 97 |
| Name | TitleLabel(1) |
| BackStyle | 0 '透明 |
| Caption | 目指せ!! |
| Font | ＭＳ Ｐゴシック |
| …Size | 18 |
| ForeColor | &H000000FF& |
| Height | 33 |
| Index | 1 |
| Left | 32 |
| TabIndex | 1 |
| Top | 7 |
| Width | 97 |
| Name | TitleLabel(2) |
| Alignment | 2 '中央揃え |
| BackStyle | 0 '透明 |
| Caption | 千切りマスター |
| Font | ＭＳ Ｐゴシック |
| …Size | 36 |
| ForeColor | &H0000FFFF& |
| Height | 57 |
| Index | 2 |
| Left | 8 |
| TabIndex | 2 |
| Top | 32 |
| Width | 313 |
| Name | TitleLabel(3) |
| Alignment | 2 '中央揃え |
| BackStyle | 0 '透明 |
| Caption | 千切りマスター |
| Font | ＭＳ Ｐゴシック |
| …Size | 36 |
| ForeColor | &H00FF0000& |
| Height | 57 |
| Index | 3 |
| Left | 8 |
| TabIndex | 3 |
| Top | 35 |
| Width | 313 |

| ピクチャボックスの設定 | |
|---|---|
| Name | Main |
| Appearance | 0 'フラット |
| AutoRedraw | -1 'True |
| BackColor | &H00FFC0C0& |
| ForeColor | &H80000008& |
| Height | 89 |
| Left | 24 |
| MousePointer | 99 'ユーザー定義 |
| ScaleHeight | 87 |
| ScaleMode | 3 'ピクセル |
| ScaleWidth | 271 |
| TabIndex | 5 |
| Top | 112 |
| Width | 273 |

| タイマーの設定 | |
|---|---|
| Name | Timer1 |
| Left | 360 |
| Top | 3120 |

| シェイプの設定 | |
|---|---|
| Name | ShapeChr(0) |
| BackStyle | 1 '不透明 |
| Height | 9 |
| Index | 0 |
| Left | 40 |
| Top | 40 |
| Visible | 0 'False |
| Width | 169 |
| Name | ShapeChr(1) |
| FillColor | &H00008000& |
| FillStyle | 0 '塗りつぶし |
| Height | 15 |
| Index | 1 |
| Left | 32 |
| Top | 40 |
| Visible | 0 'False |
| Width | 17 |
| Name | ShapeChr(2) |
| FillColor | &H0000FF00& |
| FillStyle | 0 '塗りつぶし |
| Height | 17 |
| Index | 2 |
| Left | 24 |
| Shape | 4 '丸みのある長方形 |
| Top | 32 |
| Visible | 0 'False |
| Width | 17 |
| Name | ShapeChr(3) |
| BorderStyle | 0 '透明 |
| FillColor | &H00FFFFFF& |
| FillStyle | 0 '塗りつぶし |
| Height | 41 |
| Index | 3 |
| Left | 32 |
| Shape | 2 '楕円 |
| Top | 24 |
| Visible | 0 'False |
| Width | 209 |
| Name | ShapeChr(4) |
| BackStyle | 1 '不透明 |
| FillColor | &H00008000& |
| FillStyle | 0 '塗りつぶし |
| Height | 25 |
| Index | 4 |
| Left | 8 |
| Shape | 2 '楕円 |
| Top | 24 |
| Visible | 0 'False |
| Width | 33 |
| Name | ShapeChr(5) |
| FillColor | &H00008000& |
| FillStyle | 0 '塗りつぶし |
| Height | 25 |
| Index | 5 |
| Left | 24 |
| Shape | 2 '楕円 |
| Top | 48 |
| Visible | 0 'False |
| Width | 17 |
| Name | ShapeChr(6) |
| BorderStyle | 0 '透明 |
| FillColor | &H00004000& |
| FillStyle | 0 '塗りつぶし |
| Height | 25 |
| Index | 6 |
| Left | 32 |
| Shape | 2 '楕円 |
| Top | 32 |
| Visible | 0 'False |
| Width | 9 |
| Name | ShapeChr(7) |
| FillColor | &H0000C000& |
| FillStyle | 0 '塗りつぶし |
| Height | 41 |
| Index | 7 |
| Left | 40 |
| Shape | 2 '楕円 |
| Top | 24 |
| Visible | 0 'False |
| Width | 97 |
| Name | ShapeChr(8) |
| FillColor | &H00008000& |
| FillStyle | 0 '塗りつぶし |
| Height | 25 |
| Index | 8 |
| Left | 32 |
| Shape | 2 '楕円 |
| Top | 32 |
| Visible | 0 'False |
| Width | 25 |
| Name | ShapeChr(9) |
| FillColor | &H00008000& |
| FillStyle | 0 '塗りつぶし |
| Height | 41 |
| Index | 9 |
| Left | 32 |
| Shape | 2 '楕円 |
| Top | 40 |
| Visible | 0 'False |
| Width | 97 |
| Name | ShapeChr(10) |
| FillColor | &H0000FF00& |
| FillStyle | 0 '塗りつぶし |
| Height | 41 |
| Index | 10 |
| Left | 32 |
| Shape | 2 '楕円 |
| Top | 8 |
| Visible | 0 'False |
| Width | 97 |
| Name | ShapeChr(11) |
| FillColor | &H00008000& |
| FillStyle | 0 '塗りつぶし |
| Height | 41 |
| Index | 11 |
| Left | 24 |
| Shape | 2 '楕円 |
| Top | 24 |
| Visible | 0 'False |
| Width | 65 |

| ラインの設定 | |
|---|---|
| Name | StartLine |
| BorderColor | &H80000005& |
| BorderWidth | 3 |
| X1 | 32 |
| X2 | 32 |
| Y1 | 0 |
| Y2 | 88 |
| Name | GoalLine |
| BorderColor | &H000000FF& |
| BorderWidth | 7 |
| X1 | 224 |
| X2 | 224 |
| Y1 | 0 |
| Y2 | 88 |
| Name | Line1 |
| BorderColor | &H000000FF& |
| Index | 0 |
| Visible | 0 'False |
| X1 | 232 |
| X2 | 232 |
| Y1 | 8 |
| Y2 | 80 |

★「ExcelでのBASICプログラミング」はいいですね。この記事をきっかけに僕もExcel VBA始めました。もちろん，totoで高額当選も狙います。（東京都杉並区・**ぱびる21歳**）……【**編**：「Excel～」の人気が高いですね。投稿もお待ちしていますよ。】

## 《第２表》frmAboutのプロパティ表

| | |
|---|---|
| **フォームの設定** | |
| Name | frmAbout |
| BackColor | &H00C0FFC0& |
| BorderStyle | 1 '固定(実線) |
| Caption | About |
| Height | 2895 |
| Left | 0 |
| LinkTopic | Form2 |
| MaxButton | 0 'False |
| MinButton | 0 'False |
| ScaleHeight | 2520 |
| ScaleWidth | 4680 |
| StartUpPosition | 2 '画面の中央 |
| Top | 0 |
| Width | 4770 |
| **コマンドボタンの設定** | |
| Name | Command1 |
| BackColor | &H0080FF80& |
| Caption | OK |
| Height | 375 |
| Left | 1680 |
| Style | 1 'グラフィックス |
| TabIndex | 5 |
| Top | 1920 |
| Width | 1215 |
| **ラベルの設定** | |
| Name | Label1(0) |
| BackStyle | 0 '透明 |
| Caption | 目指せ!! |
| Font | ＭＳ Ｐゴシック |
| …Size | 18 |
| ForeColor | &H0000FFFF& |
| Height | 495 |
| Index | 0 |
| Left | 360 |
| TabIndex | 3 |
| Top | 0 |
| Width | 1455 |
| Name | Label1(1) |
| BackStyle | 0 '透明 |
| Caption | 目指せ!! |
| Font | ＭＳ Ｐゴシック |
| …Size | 18 |
| ForeColor | &H000000FF& |
| Height | 495 |
| Index | 1 |
| Left | 360 |
| TabIndex | 2 |
| Top | 30 |
| Width | 1455 |
| Name | Label1(2) |
| Alignment | 2 '中央揃え |
| BackStyle | 0 '透明 |
| Caption | 千切りマスター |
| Font | ＭＳ Ｐゴシック |
| …Size | 36 |
| ForeColor | &H0000FFFF& |
| Height | 855 |
| Index | 2 |
| Left | 45 |
| TabIndex | 1 |
| Top | 510 |
| Width | 4695 |
| Name | Label1(3) |
| Alignment | 2 '中央揃え |
| BackStyle | 0 '透明 |
| Caption | 千切りマスター |
| Font | ＭＳ Ｐゴシック |
| …Size | 36 |
| ForeColor | &H00FF0000& |
| Height | 855 |
| Index | 3 |
| Left | 0 |
| TabIndex | 0 |
| Top | 480 |
| Width | 4695 |
| Name | Label2 |
| BackStyle | 0 '透明 |
| Caption | By tayosan |
| Font | ＭＳ Ｐゴシック |
| …Size | 18 |
| ForeColor | &H00FF00FF& |
| Height | 495 |
| Left | 1440 |
| TabIndex | 4 |
| Top | 1200 |
| Width | 1815 |
| **ラインの設定** | |
| Name | Line1 |
| BorderColor | &H0000FF00& |
| BorderWidth | 2 |
| X1 | 120 |
| X2 | 4560 |
| Y1 | 1800 |
| Y2 | 1800 |
| Name | Line2 |
| BorderColor | &H00008000& |
| X1 | 120 |
| X2 | 4560 |
| Y1 | 1800 |
| Y2 | 1800 |

## 《リスト１》Form1のプログラム・リスト

```
'-----------------------------------------------------------
'        目指せ!!千切りマスター
'        Copyright tayosan 2001/8/10
'-----------------------------------------------------------

'変数の宣言を義務付ける
Option Explicit

'サウンド再生用のAPI宣言
Private Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA" _
    (ByVal lpszSoundName As String, ByVal uFlags As Long) As Long

Const DivisionPoint = 30                '得点となる基準
Const MaxTime = 100                     'タイムポイントの最大値(2倍)
Const ChopMax = 100                     '切り分けられる最大

Dim ChopX(ChopMax) As Integer           '切った時のX座標
Dim ChopCount As Integer                '切った回数
Dim Score As Integer                    'スコア
Dim Hiscore(2) As Integer               'ハイスコア
Dim SelectObject As Byte                '選択している対象物
Dim ObjectWidth(2) As Integer           '対象物の長さ
Dim ObjectHard(2) As Integer            '対象物の硬さ
Dim ObjectData(2, 1) As Byte            '対象物のシェイプの番号
Dim GameFlag As Byte                    'ゲーム用フラグ
Dim CountTime As Byte                   '時間のカウント用
Dim YourTime As Byte                    '千切りにかかった時間
Dim I As Integer                        'ループ用変数

Private Sub Form_Load()
   'タイトルの設定
   App.Title = "目指せ!!千切りマスター"
   Me.Caption = App.Title
   '対象物（ゴールライン表示）の設定
   ObjectWidth(0) = 200
   ObjectWidth(1) = 230
   ObjectWidth(2) = 135
   '対処物の硬さ設定
   ObjectHard(0) = 4
   ObjectHard(1) = 2
   ObjectHard(2) = 7
   '対象物の表示用データ設定
   ObjectData(0, 0) = 0: ObjectData(0, 1) = 2
   ObjectData(1, 0) = 3: ObjectData(1, 1) = 6
   ObjectData(2, 0) = 7: ObjectData(2, 1) = 11
   SelectObject = 0
   'シェイプのラインを呼び出して一番上に設定
   Line1(0).ZOrder 0
   For I = 1 To ChopMax
      Load Line1(I)
      Line1(I).ZOrder 0
   Next
   Init
   'スコア読み込み
   On Error GoTo Err:
      Open App.Path + "¥SCORE.TXT" For Input As #1
         For I = 0 To 2
            Input #1, Hiscore(I)
         Next
      Close
   Exit Sub
```

★こんにちは，編集部のみなさん。僕の兄は専門学校でC言語を勉強し始めたようです。と同時に，僕の部屋からベーマガのバックナンバーを持ち出すようになりました。しかも返さないんです（泣）。（京都市・**ワンツー16歳**）……【**影：今月は（泣）が多いぞ。**】

```
Err:
    Close
    SaveScore
End Sub

Sub Init()
    'ピクチャ上のマウスカーソルを設定
    Main.MouseIcon = LoadPicture(App.Path + "¥包丁1.cur")
    'メニューを使用可能にする
    mnuGame.Enabled = True
    mnuObject.Enabled = True
    mnuHelp.Enabled = True
    '表示したラインを隠す
    For I = 0 To ChopMax
        Line1(I).Visible = False
    Next
    GameFlag = 0
    Timer1.Interval = 0
    YourTime = 0
    Score = 0
    ChopCount = 0
    '対象物を表示
    ShowObject
    Label1.Caption = "ゲームを開始してください。"
End Sub

Private Sub Main_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    'マウスボタンが押された
    If GameFlag = 2 Then
        'マウスカーソル変更
        Main.MouseIcon = LoadPicture(App.Path + "¥包丁2.cur")
        ChopCount = ChopCount + 1
        ChopX(ChopCount) = X - StartLine.X1
        'クリックした位置に線を表示
        Line1(ChopCount - 1).X1 = X
        Line1(ChopCount - 1).X2 = X
        Line1(ChopCount - 1).Visible = True
        'ウェーブ再生
        WavePlay "don"
    End If
End Sub

Private Sub Main_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    'マウスが動いた
    If GameFlag = 2 Then
        'マウスカーソルを元に戻す
        Main.MouseIcon = LoadPicture(App.Path + "¥包丁1.cur")
        If X > ObjectWidth(SelectObject) Then
            'ゴールを超えた
            GameFlag = 3
            CountTime = 3
            Timer1.Interval = 1000
        End If
    End If
End Sub

Private Sub mnuEnd_Click()
    End
End Sub

Private Sub mnuGameStart_Click()
    'メニューの開始が押された
    WavePlay "GameStart"
    mnuGame.Enabled = False
    mnuObject.Enabled = False
    mnuHelp.Enabled = False
    GameFlag = 1
    CountTime = 4
    Label1.Caption = "Are you ready?"
    Timer1.Interval = 1000
End Sub

Private Sub mnuHelpOther_Click()
    'メニューでその他を押した
    WavePlay "don"
    frmAbout.Show
    Me.Hide
End Sub

Private Sub mnuHiscoreDelete_Click()
    'ハイスコア初期化
    If MsgBox("ハイスコアを初期化します。" + vbNewLine + _
        "よろしいですか？", vbYesNo + vbQuestion) = vbYes Then
        For I = 0 To 2
            Hiscore(I) = 0
        Next
        SaveScore
    End If
End Sub

Private Sub mnuHiscoreShow_Click()
    'ハイスコアの表示
    Dim ST As String

    WavePlay "pp"
    For I = 0 To 2
        ST = ST + Right("    " + mnuObjectSelect(I).Caption, 4) + _
            " >Hi Score..." + Str(Hiscore(I)) + String(2, vbNewLine)
    Next
    MsgBox "-- Hi score --" + String(3, vbNewLine) + ST
    WavePlay "kuru"
End Sub

Private Sub mnuObjectSelect_Click(Index As Integer)
    '対象物の変更
    SelectObject = Index
    For I = 0 To 2
        mnuObjectSelect(I).Checked = False
        If I = Index Then mnuObjectSelect(I).Checked = True
    Next
    ShowObject
    WavePlay "kuru"
End Sub

Sub ShowLine()
    'バックのガイド表示
    Me.Refresh
    For I = 0 To Main.Width Step 30
        Main.Line (I, 0)-(I, Main.Height), QBColor(15)
    Next
End Sub

Sub ShowObject()
    '対象物の表示
    ShowLine
    For I = 0 To 11
```



★右ページ側に載るといいんですけど…。最近のマイブームは畑仕事だったりします。電話機の子機で携帯気分（哀）。私は，ぴっしり書いてしまうんです。A型なもので。（鹿児島県川内市・**ちゃっぴい28歳**）……**【つぐ美：もっと書いてありましたが省略しました。】**

```
        If ObjectData(SelectObject, 0) <= I And _
            ObjectData(SelectObject, 1) >= I Then
            ShapeChr(I).Visible = True
        Else
            ShapeChr(I).Visible = False
        End If
    Next
    'ゴール位置の変更
    GoalLine.X1 = ObjectWidth(SelectObject)
    GoalLine.X2 = ObjectWidth(SelectObject)
End Sub

Private Sub Timer1_Timer()
    Select Case GameFlag
        Case 1
            '始めのカウント
            WavePlay "don"
            CountTime = CountTime - 1
            Label1.Caption = CountTime
            If CountTime = 0 Then
                GameFlag = 2
                Timer1.Interval = 100
                Label1.Caption = "GO!!"
            End If
        Case 2
            'ゲーム中のタイム表示
            If YourTime >= 200 Then
                WavePlay "bu"
                Label1.Caption = "制限時間を超過しました!"
                GameFlag = 4
                CountTime = 6
                Timer1.Interval = 1000
                Exit Sub
            End If
            YourTime = YourTime + 1
            Label1.Caption = Str(YourTime / 10)
        Case 3
            'ゲーム終了後のウェイト
            CountTime = CountTime - 1
            If CountTime = 0 Then
                ScoreShow
                GameFlag = 4
                CountTime = 6
            End If
        Case 4
            'スコア表示後のウェイト
            CountTime = CountTime - 1
            If CountTime = 0 Then
                Init
            End If
    End Select
End Sub

Sub ScoreShow()
    'スコアの計算と表示
    Dim ChopWidth As Integer
    Dim PL As Integer
    Dim ST As String
    Dim MinWidth As Integer
    Dim MaxWidth As Integer

    MinWidth = Main.ScaleWidth
    MaxWidth = 0
    For I = 1 To ChopCount
        '千切りの幅を格納
        ChopWidth = ChopX(I) - ChopX(I - 1)
        '一番広い幅と、一番狭い幅を調べる
        If ChopWidth < MinWidth Then MinWidth = ChopWidth
        If ChopWidth > MaxWidth Then MaxWidth = ChopWidth
        If ObjectHard(SelectObject) > ChopWidth Then
            '千切り失敗（潰れた・・・）
            PL = -((ObjectHard(SelectObject) - ChopWidth) ^ 2)
        Else
            '得点になる（DivisionPointより切った幅が大きかったら減点
            PL = DivisionPoint - ChopWidth
        End If
        'スコアに加算
        Score = Score + PL
    Next
    '様々な値を最終的に格納する
    Score = (Score + (MaxTime - YourTime) * 2 - (MinWidth + MaxWidth))
    If Hiscore(SelectObject) < Score Then
        'ハイスコア！
        WavePlay "GameEnd"
        Hiscore(SelectObject) = Score
        ST = " | Hiscore!!"
        SaveScore
    Else
        WavePlay "gyu"
    End If
    Label1.Caption = "Score=" + Str(Score) + ST
End Sub

Private Sub WavePlay(Wname As String)
    'WAVを再生
    sndPlaySound Wname, 1
End Sub

Private Sub WaveClose()
    'WAVの再生を止める
    sndPlaySound vbNullString, 1
End Sub

Sub SaveScore()
    'ハイスコアの保存
    Open App.Path + "¥SCORE.TXT" For Output As #1
        For I = 0 To 2
            Print #1, Hiscore(I)
        Next
    Close
End Sub
```

**《リスト２》frmAboutのプログラム・リスト**

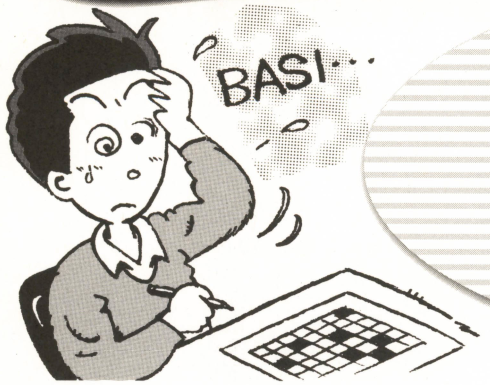
```
Private Sub Command1_Click()
    Form1.Show
    Unload frmAbout
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Command1_Click
End Sub
```



★さてと，そろそろ再開するかな…。（長野県松本市・TAC25歳）……【**影**：さいかい（く，苦しい）。**編**：再会できてうれしいです。**影**：この動物なに？サイかいってか。**Dr.D**：編集部ボーリング大会ではワシは最下位じゃった。**つぐ美**：なに遊んでるんですか！！】

# Windows(Visual Basic V6.0)用




### スケルトンパズルを瞬時に解答

実用

# スケルトン

阿部　雅之（42歳）

## プログラムの内容

「スケルトン」というクロスワードパズルをご存知でしょうか。後ほど簡単な例をあげて説明しますが，縦横に決められた字数の枠があり，そこに決められた単語群から字数と交差状態を判断して，クロスワードを埋めるものです。

クロスワードといえば「ナンクロ」が有名ですが，この場合には埋める語群が決まっていないため，コンピュータに解かせることは至難の技です。

しかし，「スケルトン」の場合には文字単位ではなく，単語単位で決定できるために，それほど難しくはないのです。そこで，コンピュータに解かせるプログラムを作ってみました。

## プログラムの使用方法

まず問題を入力しなければ，はじまりませんから，**リスト2**のような形式で問題のデータの入力ファイルを作ります。ファイルの拡張子はtxtでかまいません。

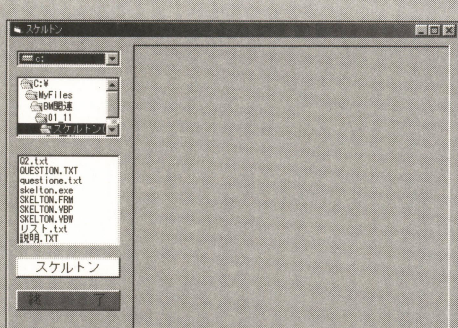
**リスト2**は，①単語の個数（半角），②埋める単語群，③埋める図形(半角)の順に並べ，順序を変更したり，省略はできません。埋める単語群は必ずしも文字数の少ない順序で並べる必要はありません。

最後の埋める図形は，それを見た目で左上を基準に，単語を埋めるマスがあれば1，なければ0を半角で指定します。

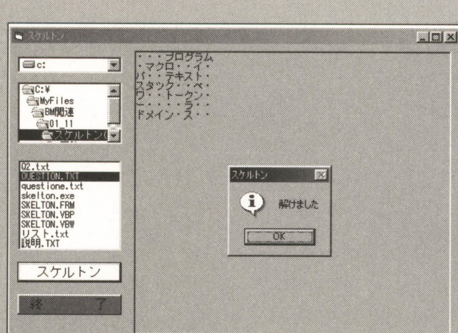
なお，途中に空白が入っていたり，何も入っていない行があってもかまわないようにして，入力不慣れに備えて利用者本位にしたところも，このプログラムの大きな特長となっています。たとえば，「マクロ」の部分が「マ△ク△ロ」になっていても，改行マークだけの行や，空白（全角半角は問いません）ばかりの行が混じっていても，それらをすべて読みとばすように作ってあります（プログラムを作るとき，正常な場合よりもエラー処理のほうが大変です）。

## 問題にエラーがあったとき

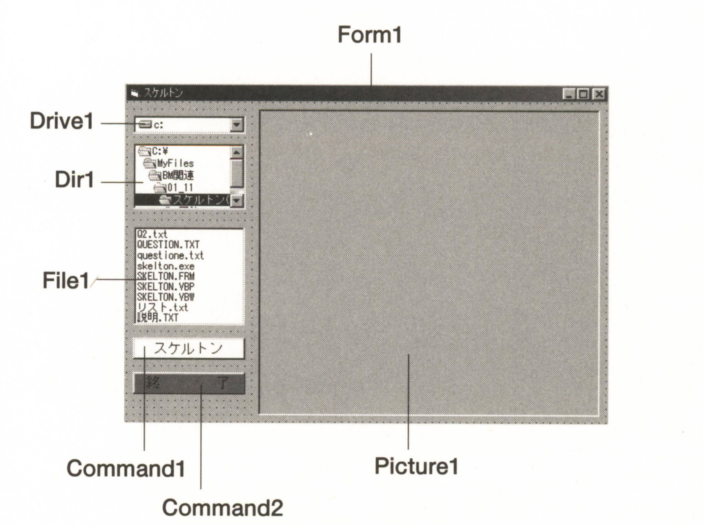
解けなかった場合には「手詰まりです」と表示されるので，問題の入力誤りがな




《写真1》記述した問題ファイルを読み込めば




《写真2》一瞬で解答を表示してくれる




フォームの配置図



★「影さんのマジックボックス」って，影さんがいなくても成立すると思っているのは僕だけでしょうか。（京都市・Essa14歳）……【影：とんでもない！私がいなきゃ始まらないのだ。つぐ美：だから掲載されたりされなかったり？影：うぐっ！投稿待ってます。】

## CHECKER FLAG

**影**：阿部さんお得意の，パズルの解答を求めるソフト。これでまた，パズル雑誌を買ってきて，このソフトを使えば，すぐにプレゼントに応募できるぜー。

**Dr.D**：影クンみたいに割り切って考える人もいれば，中には「こういうパズルは，解答を求めるまでの行程が面白いので，その解答を求めるソフトは邪道」という人もいると思う。じゃが，プログラム面やアルゴリズムの面で見ると，レベルの高いソフトであることは，認識してほしいものじゃ。とくにこのプログラムは，問題テキストの読み込み部分に工夫が見られ，使い勝手の良い形になっておる。

**つぐ美**：説明文にも書いてあるけど，瞬時に解答が求まるのは，チョット した爽快感があるわね。

いかをまず見てください。クロスワードでは，促音（“っ”）は使いません。不幸にして無限ループへ入ってしまった場合に備えて終了コマンドを，非常停止コマンドとして使用できるようになっています。

## プログラムのしくみ

基本的には，文字数の一致，交差部分の一致，そしてすでに単語の一部が埋まっているものはそれを優先して検査する，という方法で解いています。同時に，確定した部分を再度検査しないように，使用済み判定フラグを立てて対応しています。

プログラムの世界では計算処理にとって変わった感のある，文字列処理のアルゴリズムが終始一貫して使われています。もしC言語で作ればポインタばかりの大変な（？）プログラムになることでしょう。

ところがこれで完成と思っていたところ，重要な見落としに気づきました。「スケルトン」には，単語を埋める問題ばかりではなく，埋まらない単語を探す問題もあるのです。そこで，単語の数を自動判別させて，残った単語を表示させる処理を最後の部分に追加しました。ベーマガ8月号本欄「少し時間をおいて再び検査をする」ことの重要性を改めて納得させられた重大なバグでした。

**《第1表》プロパティ表**

| フォームの設定 | |
|---|---|
| Name | Form1 |
| Caption | スケルトン |
| Font | ＭＳ ゴシック |
| …Size | 10 |
| Height | 5955 |
| Left | 0 |
| LinkTopic | Form1 |
| ScaleHeight | '5550 |
| ScaleWidth | '8310 |
| StartUpPosition | 3 'Windows の既定値 |
| Top | 0 |
| Width | 8430 |
| **ピクチャボックスの設定** | |
| Name | Picture1 |
| Font | ＭＳ ゴシック |
| …Size | 9 |
| Height | 5295 |
| Left | 2280 |
| ScaleHeight | 5235 |
| ScaleWidth | 5835 |
| TabIndex | 5 |
| Top | 120 |
| Width | 5895 |
| **コマンドボタンの設定** | |
| Nmae | Command1 |
| BackColor | &H80000009& |
| Caption | スケルトン |
| Font | ＭＳ ゴシック |
| …Size | 12 |
| Height | 375 |
| Left | 120 |
| Style | 1 'グラフィックス |
| TabIndex | 1 |
| Top | 4080 |
| Width | 1935 |
| Name | Command2 |
| BackColor | &H000000FF& |
| Caption | 終　　了 |
| Font | ＭＳ ゴシック |
| …Size | 12 |
| Height | 375 |
| Left | 120 |
| MaskColor | &H80000004& |
| Style | 1 'グラフィックス |
| TabIndex | 4 |
| Top | 4680 |
| Width | 1935 |
| **ドライブリストボックスの設定** | |
| Name | Drive1 |
| Font | ＭＳ ゴシック |
| …Size | 9 |
| Height | 300 |
| Left | 120 |
| TabIndex | 0 |
| Top | 240 |
| Width | 1935 |
| **ディレクトリリストボックスの設定** | |
| Name | Dir1 |
| Font | ＭＳ ゴシック |
| …Size | 9 |
| Height | 1140 |
| Left | 120 |
| TabIndex | 3 |
| Top | 720 |
| Width | 1935 |
| **ファイルリストボックスの設定** | |
| Name | File1 |
| Font | ＭＳ ゴシック |
| …Size | 9 |
| Height | 1710 |
| Left | 120 |
| TabIndex | 2 |
| Top | 2160 |
| Width | 1935 |

**《第2表》変数表**

| | |
|---|---|
| nmax | 単語の最大許容数（問題によって変更します） |
| n | 単語の数 |
| area_max | 図形中の埋めるべき言葉の数 |
| ntate | 図形の縦のマス数 |
| nyoko | 図形の横のマス数 |
| ncount | 図形中の埋めるべき言葉の残りの数 |
| mcount | 左右方向の図形中の埋めるべき言葉の残りの数 |
| word_max | 単語の最大文字数 |
| word_min | 単語の最小文字数 |
| moji() | 単語格納領域 |
| matrix() | 図形中の確定した単語や文字の格納領域 |
| x_start() | 図形中の上下方向の始点座標 |
| x_fin() | 図形中の上下方向の終点座標 |
| y_start() | 図形中の左右方向の始点座標 |
| y_fin() | 図形中の左右方向の終点座標 |
| d_direct() | 上下方向か左右方向を区別するフラグ |
| c_count() | 図形中の埋める文字数 |
| fpat() | 図形パターン |
| w_count() | 何文字の単語がいくつあるかを数えるカウンタ |
| moji_len() | 各単語の文字数 |
| moji_fin() | 使用済み単語の判定フラグ |
| word() | 図形中で埋まった単語の一時的格納領域 |
| word_fin() | 図形中で埋まった領域の判定フラグ |

## 最後に

ほとんどの問題をわずか数秒以内で解くことができますが，限界もあります。

まず，背理法といって「あるマスにある単語を仮定すると，手詰まりが発生するので，この単語は入れることができない」という一種の消去法には対応していないということです。この「背理法」を用いなければ解けない難問もわずかながらあります。

つぎに，単語ベースで解く「言葉」の問題を対象としており，一文字単位で埋めなければならない，数字の問題には対応していないということです。このあたりが，今後改造の課題となるでしょう。

いずれにしても，埋めるべき単語の数が200個近くある問題の答えが，一瞬にしてパソコンの画面に表示されたときの気分は格別です。

**《リスト1》スケルトンのプログラム・リスト**

```
Option Explicit
Rem 単語数に合わせて大きくします
Const nmax = 150
Const soldinit = 0
Const soldout = 1
Rem 全角のスペースです
Const blank = "　"
Rem 収束判定フラグ
Dim retcode As Integer, precode As Integer
Rem アドレス
Dim n As Integer, ntate As Integer, nyoko As Integer, area_max As Integer
Rem 縦横のマスの数
Dim ncount As Integer, mcount As Integer
Rem 図形領域
Dim moji(nmax) As String, matrix(nmax) As String, fpat(nmax) As String
Rem 整理領域
Dim x_start(nmax) As Integer, y_start(nmax) As Integer
Dim x_fin(nmax) As Integer, y_fin(nmax) As Integer
```

```
Dim d_direct(nmax) As Integer, c_count(nmax) As Integer
Rem 単語の情報－文字数・使用状況など
Dim w_count(nmax) As Integer, word_max As Integer, word_min As Integer
Dim moji_len(nmax) As Integer, moji_fin(nmax) As Integer
Dim word(nmax) As String, word_fin(nmax) As Integer
Function spclear(inmoji As String)
  Rem 文字間の余分な空白を除くプロシジャ
  Dim i As Integer
  Dim outmoji As String
  outmoji = ""
  For i = 1 To Len(inmoji)
    If Mid(inmoji, i, 1) <> " " And Mid(inmoji, i, 1) <> blank Then
      outmoji = outmoji & Mid(inmoji, i, 1)
    End If
  Next i
  spclear = outmoji
End Function
Private Sub list()
  Rem 文字表示のプロシジャ
  Dim i As Integer, j As Integer
  Picture1.Cls
  For i = 1 To ntate
    For j = 1 To Len(matrix(i))
      If Mid(matrix(i), j, 1) = "*" Then
        Picture1.Print "・";
      Else
        Picture1.Print Mid(matrix(i), j, 1);
      End If
    Next j
    Picture1.Print
  Next i
End Sub
Private Sub limit()
  Rem 収束判定のプロシジャ
  Dim i As Integer
  retcode = 0
  For i = 1 To area_max
    retcode = retcode + word_fin(i)
  Next i
End Sub
Private Sub extra()
  Rem 残った単語を探すプロシジャ
  Dim i As Integer
  Picture1.Print "残った単語は"
  For i = 1 To n
    If moji_fin(i) = soldinit Then
      Picture1.Print moji(i)
    End If
  Next i
End Sub
Private Sub flaginit()
  Rem 各種情報フラグの初期化プロシジャ
  Dim i As Integer, j As Integer
  Dim fmt As String
  Rem 単語関連
  For i = 1 To nmax
    moji_fin(i) = soldinit
    word_fin(i) = soldinit
  Next i
  Rem 表示図形
  For i = 1 To ntate
    fmt = ""
    For j = 1 To Len(fpat(i))
      If Mid(fpat(i), j, 1) = "0" Then
        fmt = fmt & "*"
      Else
        fmt = fmt & blank
      End If
    Next j
    matrix(i) = fmt
  Next i
```

```
  Rem 収束判定フラグ
  retcode = soldinit
  precode = -1
  area_max = 1
End Sub
Private Sub dstore(x1 As Integer, x2 As Integer, y1 As Integer, y2 As Integer)
  Rem 図形中の単語の座標格納のプロシジャ
  x_start(ncount) = x1
  y_start(ncount) = y1
  x_fin(ncount) = x2
  y_fin(ncount) = y2
End Sub
Private Sub wordcnt()
  Rem 単語の文字数と出現回数を数えるプロシジャ
  Dim i As Integer
  For i = 1 To nmax
    w_count(i) = 0
  Next i
  For i = 1 To n
    If moji_fin(i) = soldinit Then
      w_count(Len(moji(i))) = w_count(Len(moji(i))) + 1
    End If
  Next i
End Sub
Private Sub faulone()
  Rem 一意に決まる単語を埋めるプロシジャ
  Dim i As Integer, j As Integer, k As Integer
  For i = word_min To word_max
    If w_count(i) = 1 Then
      For j = 1 To n
        If i = moji_len(j) And moji_fin(j) = soldinit Then
          For k = 1 To ncount
            If c_count(k) = i And word_fin(k) = soldinit Then
              word(k) = moji(j)
              word_fin(k) = soldout
              moji_fin(j) = soldout
              Exit For
            End If
          Next k
        End If
      Next j
    End If
  Next i
End Sub
Private Sub wildcard()
  Rem 一部が埋まった単語走査のプロシジャ
  Dim spflag As Integer, twinflag As Integer, rcount As Integer
  Dim spaddr(nmax) As Integer, fauladdr As Integer
  Dim i As Integer, j As Integer, k As Integer
  For i = 1 To ncount
    If word_fin(i) = soldinit Then
      spflag = 0
      For j = 1 To Len(word(i))
        If Mid(word(i), j, 1) <> blank Then
          spflag = 1
          Exit For
        End If
      Next j
      If spflag = 1 Then
        For j = 1 To nmax
          spaddr(j) = 0
        Next j
        For j = 1 To Len(word(i))
          If Mid(word(i), j, 1) <> blank Then
            spaddr(j) = 1
          End If
        Next j
        rcount = 0
        For k = 1 To n
          If c_count(i) = moji_len(k) And moji_fin(k) = soldinit Then
            twinflag = 0
```



★去年といい，今年といい，私にとって11月号はある意味，鬼門のようです。(青森県南津軽郡・ていこく22歳) ……【**編**：？？？どういう意味でしょうねぇ。**影**：さすがの私も悩んでしまうぞ。この疑問を早く解決しないと，寝不足になってしまうー。】

```
            For j = 1 To Len(moji(k))
              If spaddr(j) = 1 Then
                If Mid(word(i), j, 1) = Mid(moji(k), j, 1) Then
                  twinflag = 1
                Else
                  twinflag = 0
                  Exit For
                End If
              End If
            Next j
            If twinflag = 1 Then
              rcount = rcount + 1
              fauladdr = k
            End If
          End If
        Next k
        If rcount = 1 Then
          word(i) = moji(fauladdr)
          word_fin(i) = soldout
          moji_fin(fauladdr) = soldout
        End If
      End If
    End If
  Next i
End Sub
Private Sub fmtcnv()
  Rem 図形内文字走査のプロシジャ
  Dim i As Integer, j As Integer, k As Integer
  Dim flag As Integer, gcount As Integer
  Dim wdat As String
  ncount = 0
  Rem 横方向
  For i = 1 To ntate
    flag = 0
    wdat = matrix(i)
    gcount = 0
    For j = 1 To Len(wdat)
      If Mid(wdat, j, 1) = "*" Then
        If flag = 1 And gcount > 1 Then
          flag = 0
          ncount = ncount + 1
          Call dstore(i, i, j - gcount, j - 1)
          word(ncount) = ""
          For k = y_start(ncount) To y_fin(ncount)
            word(ncount) = word(ncount) & Mid(wdat, k, 1)
          Next k
          gcount = 0
      Else
        flag = 0
        gcount = 0
      End If
    ElseIf Mid(wdat, j, 1) <> "*" Then
       gcount = gcount + 1
       If flag = 0 Then
         flag = 1
       End If
    End If
  Next j
  If flag = 1 And gcount > 1 Then
    ncount = ncount + 1
    Call dstore(i, i, j - gcount, Len(wdat))
    word(ncount) = ""
    For k = y_start(ncount) To y_fin(ncount)
      word(ncount) = word(ncount) & Mid(wdat, k, 1)
    Next k
  End If
Next i
For i = 1 To mcount - 1
  d_direct(i) = 0
  For j = 1 To Len(word(i))
    If Mid(word(i), j, 1) = blank Then
      d_direct(i) = 1
      Exit For
    End If
  Next j
  c_count(i) = y_fin(i) - y_start(i) + 1
Next i
mcount = ncount + 1
Rem 縦方向
For j = 1 To nyoko
  flag = 0
  wdat = ""
  For i = 1 To ntate
    wdat = wdat & Mid(matrix(i), j, 1)
  Next i
  gcount = 0
  For i = 1 To Len(wdat)
    If Mid(wdat, i, 1) = "*" Then
      If flag = 1 And gcount > 1 Then
        flag = 0
        ncount = ncount + 1
        Call dstore(i - gcount, i - 1, j, j)
        word(ncount) = ""
        For k = x_start(ncount) To x_fin(ncount)
          word(ncount) = word(ncount) & Mid(wdat, k, 1)
        Next k
        gcount = 0
      Else
        flag = 0
        gcount = 0
      End If
    ElseIf Mid(wdat, i, 1) <> "*" Then
       gcount = gcount + 1
       If flag = 0 Then
         flag = 1
       End If
     End If
   Next i
   If flag = 1 And gcount > 1 Then
     ncount = ncount + 1
     Call dstore(i - gcount, Len(wdat), j, j)
     word(ncount) = ""
     For k = x_start(ncount) To x_fin(ncount)
       word(ncount) = word(ncount) & Mid(wdat, k, 1)
     Next k
   End If
  Next j
  For j = mcount To ncount
    d_direct(j) = 0
    For i = 1 To Len(word(j))
      If Mid(word(j), i, 1) = blank Then
        d_direct(j) = -1
        Exit For
      End If
    Next i
    c_count(j) = x_fin(j) - x_start(j) + 1
  Next j
  If area_max < ncount Then
    area_max = ncount
  End If
End Sub
Private Sub rvscnv()
  Rem 図形展開のプロシジャ
  Dim i As Integer, j As Integer, k As Integer
  For i = 1 To ncount
    If d_direct(i) = 1 Then
      Rem 横方向
      k = 1
      For j = y_start(i) To y_fin(i)
        If Mid(matrix(x_start(i)), j, 1) = blank Then
          Mid(matrix(x_start(i)), j, 1) = Mid(word(i), k, 1)
        End If
```

```
        k = k + 1
      Next j
    ElseIf d_direct(i) = -1 Then
      Rem 縦方向
      k = 1
      For j = x_start(i) To x_fin(i)
        If Mid(matrix(j), y_start(i), 1) = blank Then
          Mid(matrix(j), y_start(i), 1) = Mid(word(i), k, 1)
        End If
        k = k + 1
      Next j
    End If
  Next i
End Sub
Private Sub cross()
  Rem 単語の交差状態を調べるプロシジャ
  Dim tate_addr As Integer, yoko_addr As Integer, i_tate As Integer, j_yoko As Integer
  Dim rcount As Integer
  Dim i As Integer, j As Integer, k As Integer, ii As Integer, jj As Integer
  For i = 1 To mcount - 1
    If word_fin(i) = soldinit Then
      For j = mcount To ncount
        If y_start(j) >= y_start(i) And y_start(j) <= y_fin(i) _
          And x_start(j) <= x_start(i) And x_start(i) <= x_fin(j) _
          And word_fin(j) = soldinit Then
          For k = y_start(i) To y_fin(i)
            If k = y_start(j) Then
              yoko_addr = k - y_start(i) + 1
              tate_addr = x_start(i) - x_start(j) + 1
              rcount = 0
              For ii = 1 To n
                If moji_fin(ii) = soldinit And c_count(i) = moji_len(ii) Then
                  For jj = 1 To n
                    If moji_fin(jj) = soldinit And c_count(j) = moji_len(jj) And _
                    Mid(moji(ii), yoko_addr, 1) = Mid(moji(jj), tate_addr, 1) And _
                    ii <> jj Then
                      rcount = rcount + 1
                      i_tate = jj
                      j_yoko = ii
                    End If
                  Next jj
                End If
              Next ii
              If rcount = 1 Then
                word(i) = moji(j_yoko)
                word(j) = moji(i_tate)
                word_fin(i) = soldout
                word_fin(j) = soldout
                moji_fin(i_tate) = soldout
                moji_fin(j_yoko) = soldout
                Exit Sub
              End If
            End If
          Next k
        End If
      Next j
    End If
  Next i
End Sub
Private Sub Command1_Click()
  Dim i As Integer, j As Integer
  Dim infile As String
  Dim ldat As String, wkdat As String
  Rem 読む
  infile = FreeFile
  If Right(File1.Path, 1) = "¥" Then
    Open File1.Path & File1.FileName For Input As #infile
  Else
    Open File1.Path & "¥" & File1.FileName For Input As #infile
  End If
  Rem 単語の数
  Input #infile, n
  Rem 単語の文字情報
  word_max = 0
  word_min = nmax
  i = 1
  Do While i <= n
    Input #infile, ldat
    If Len(ldat) > 0 Then
      wkdat = spclear(ldat)
      If Len(wkdat) > 0 Then
        moji(i) = wkdat
        moji_len(i) = Len(moji(i))
        If moji_len(i) > word_max Then
          word_max = moji_len(i)
        End If
        If moji_len(i) < word_min Then
          word_min = moji_len(i)
        End If
        i = i + 1
      End If
    End If
  Loop
  Rem 図形
  ntate = 0
  Do While Not (EOF(infile))
    Input #infile, ldat
    If Len(ldat) > 0 Then
      wkdat = spclear(ldat)
      If Len(wkdat) > 0 Then
        ntate = ntate + 1
        fpat(ntate) = ldat
      End If
    End If
  Loop
  nyoko = Len(fpat(ntate))
  Close #infile
  Call flaginit
  Do While retcode < area_max
    DoEvents
    Call wordcnt
    Call fmtcnv
    Call faulone
    Call wildcard
    Call cross
    Call rvscnv
    Call list
    Call limit
    If precode = retcode Then
      MsgBox "手詰まりです", 32, "スケルトン"
      Exit Sub
    Else
      precode = retcode
    End If
  Loop
  If n > area_max Then
    Call extra
  End If
  MsgBox "解けました", 64, "スケルトン"
End Sub
Private Sub Command2_Click()
  End
End Sub
Private Sub Dir1_Change()
  File1.Path = Dir1.Path
End Sub
Private Sub Drive1_Change()
  Dir1.Path = Drive1.Drive
End Sub
```

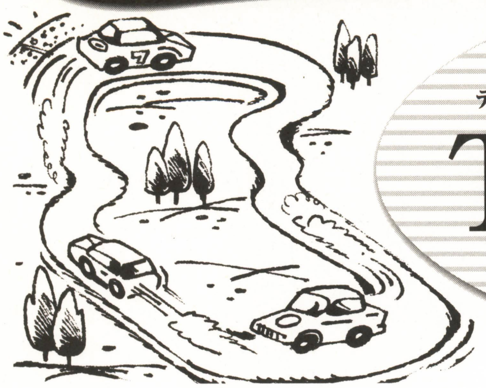
**《リスト2》スケルトン問題のリスト例**

```
10
クラス
マクロ
スタツク
テキスト
トークン
ドメイン
パスワード
プログラム
プロテクト
ライトペン
00011111
01110010
10011110
11110010
10011110
10000100
11110100
```



★3連休なんて，あっという間。（茨城県古河市・**年齢不相応41歳**）……【**影**：編集部には連休なんてありません。読者のために日夜がんばっています。**つぐ美**：あれ？影さんの場合，昼寝の時間を考えたら，3連休どころじゃなくなるはずですけど。**影**：シーッ。】

# Windows(Visual Basic V5.0)用



ライン取りが重要なレースゲーム

# T-Racer for BM

（ティー　レーサー　フォー　ビーエム）

ていこく（22歳）

## ゲームの内容

見下ろし型のレースゲームです。自機はシンプルな三角形です。

## 入力方法

(1)VBを起動し，フォームにオブジェクトを貼り付けます。

(2)Form1のリスト，Module1のリストをそれぞれ入力します。

(3)VBのメニューから「プロジェクト」→「Project1のプロパティ」を選択し，「全般」の「スタートアップの設定」を「Sub Main」に変更して，保存します。

(4)コースデータ「c1.txt」「c2.txt」「c3.txt」を入力，本体と同じフォルダに保存します。

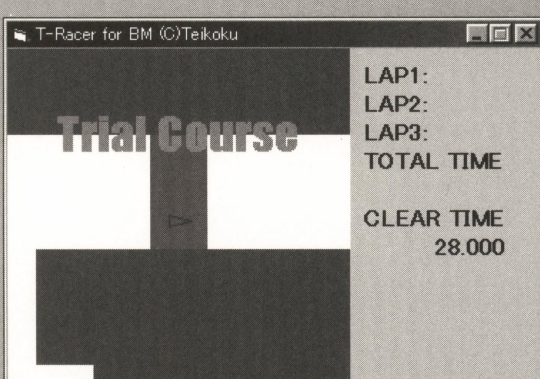
(5)一度VBを終了し，再び起動し読み込んで，実行してください。

## 遊びかた

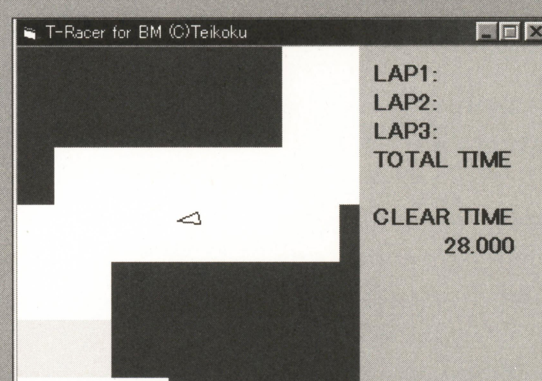
タイトル画面でスペース・キーを押すと，モード選択画面に移動します。カーソル・キーの左右で，GrandPrixTimeTrialを選択し，スペース・キーで決定します。

**・Grand Prixモード**

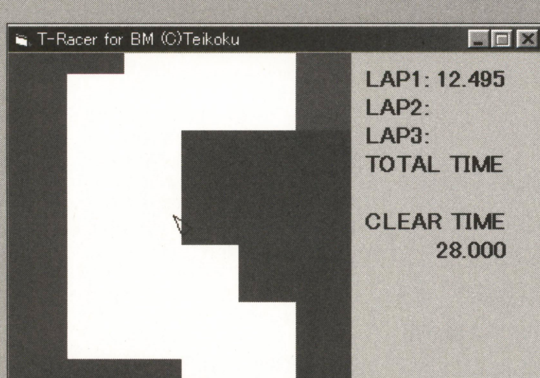
モード選択でGrand Prixを選択する



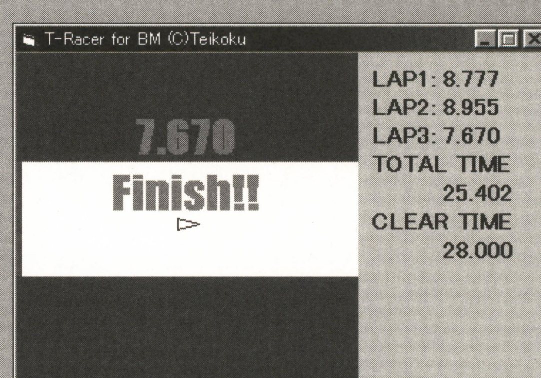

《写真１》単純なレースだが，内容はバッチリ




《写真２》コーナリングがタイムの決め手になる




《写真３》インを狙いすぎるとコースに引っかかる




《写真４》最速ラップを目指してプレイだ

★クマイヌ型の新AIBO，かわいいですね。名前もどことなくおいしそう。（北海道札幌市・春陽19歳）……【影：11月号で紹介してますが，その取材時にじつは足の裏の肉球を食べてみました。たしかにおいしかった。編：い，いつの間に！影：あの味が忘れられん。】

**CHECKER FLAG**

**つぐ美**：N-88BASICでのレースゲームのサンプルは，99年２月号(創刊200号)の「Get&Run」に掲載されています。だいぶ前の号だけど，興味のある読者は，バックナンバー購入などでチェックしてみてね。

**影**：市販のレースゲームは，もはや3Dが当然という感じだけど，こういった見下ろし型のレースゲームにも，独特のおもしろ味があるゾ。

**Dr.D**：レースゲームでメインとなる，スクロール処理は，PrintMyChr( )内のBmpCopy命令で，コースをずらしながら描画するという形じゃ。また，コースと芝の判定は，画面の色で行っているため，RaceMain( )内にある“If p = QBColor(2) Then s = s / 2”という文をコメント化すれば，芝に入ってもスピードが落ちなくなる。ゲーム性は失われるが，こういった改造ができるのもプログラムの特徴の一つなので，興味のある読者はぜひとも試してみてほしい。

と,ステージクリア型のGrandPrixモードになります。

ステージの名前とクリアタイムが表示された後，レース画面に移動します。

「Ready...」のあとにシグナル音がなり,「Go!」が出るとレースがはじまります。自機の三角形は，カーソル・キーの左右で進行方向を変え，スペース・キーでスピードを上げます。スペース・キーを離せば自然に減速します(三角形の鋭角が進行方向を示しています)。コース(白)を通ってください。芝(緑)に入ると，スピードが大幅に減速します。

また，コースには，黄色，水色，赤の三つの場所があります。これはチェックポイントです。黄色→水色→赤の順番に通らないと，１周したとはみなされません。正しい順番に通ると，音が鳴ります。

コースを１周するたびに，ラップタイムを右のほうに表示していきます。３周するとゲームが終了し，総合タイムを表示します。このとき，総合タイムがクリアタイムを下回っていたら,ステージクリアで，つぎのステージに進みます。３ステージクリアでタイトル画面に戻ります。

もちろん，クリアタイムを上回っている場合はゲーム・オーバーとなり，タイトル画面に戻ります。

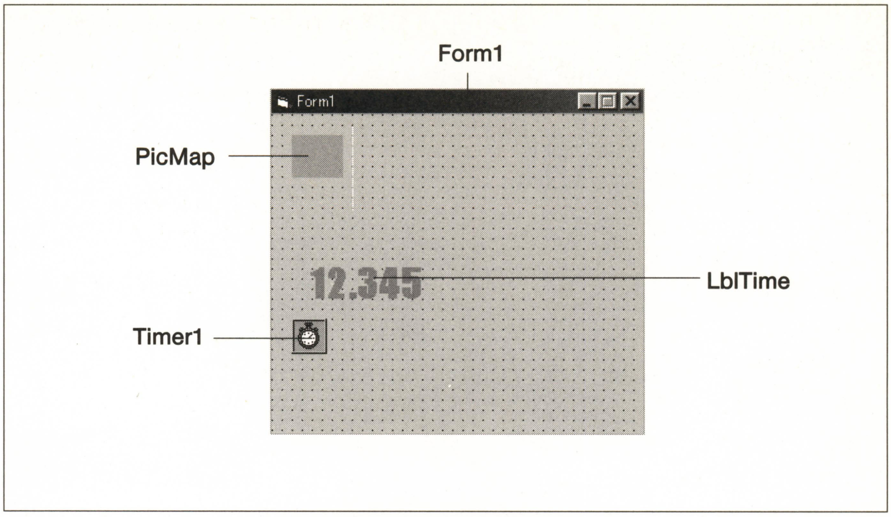
なお，コースからあまりにも外れた場合，画面に「Course Out!!」と表示され，強制的にゲームが終了しますので，注意してください。

**・Time Trialモード**

モード選択でTime Trialを選択すると，Time Trialモードになります。

コース選択画面が現れるので，カーソル・キーの左右でコースを選択し，スペース・キーで決定します。レース画面は，Grand Prixモードと同じです。

３周するとゲーム終了し，総合タイムを表示します。しばらくするとタイトル画面に戻ります。




フォームの配置図

**《第１表》プロパティ表**

| | |
|---|---|
| **フォームの設定** | |
| Name | Form1 |
| AutoRedraw | -1 'True |
| BackColor | &H0080FF80& |
| BorderStyle | 1 '固定(実線) |
| Caption | Form1 |
| Height | 4140 |
| Left | 0 |
| Font | ＭＳ Ｐゴシック |
| …Size | 12 |
| LinkTopic | Form1 |
| MaxButton | 0 'False |
| ScaleHeight | 251 |
| ScaleMode | 3 'ピクセル |
| ScaleWidth | 296 |
| StartUpPosition | 2 '画面の中央 |
| Top | 0 |
| Width | 4530 |
| **ピクチャボックスの設定** | |
| Name | PicMap |
| AutoRedraw | -1 'True |
| BorderStyle | 0 'なし |
| Font | ＭＳ Ｐゴシック |
| …Size | 9 |
| Height | 495 |
| Left | 240 |
| ScaleHeight | 33 |
| ScaleMode | 3 'ピクセル |
| ScaleWidth | 41 |
| TabIndex | 0 |
| Top | 240 |
| Visible | 0 'False |
| Width | 615 |
| **ラベルの設定** | |
| Name | LblTime |
| Alignment | 2 '中央揃え |
| AutoSize | -1 'True |
| BackStyle | 0 '透明 |
| Caption | 12.345 |
| Font | Impact |
| …Size | 24 |
| ForeColor | &H000080FF& |
| Height | 585 |
| Left | 450 |
| TabIndex | 1 |
| Top | 1680 |
| Visible | 0 'False |
| Width | 1395 |
| **タイマーの設定** | |
| Name | Timer1 |
| Enabled | 0 'False |
| Interval | 2000 |
| Left | 240 |
| Top | 2400 |

**《第２表》変数表**

| | |
|---|---|
| x,y | マップ上での自機のX座標,Y座標 |
| r | 自機の角度(0-360) |
| s | 自機のスピード |
| Chk | チェックポイント通過フラグ |
| Lap | 現在の周回数 |
| Lt(2) | 各周のラップタイム |
| tt | 総合タイム |
| Cs | 選択されたコース番号 |
| Cs_file(n) | コースデータのファイル名 |
| Cs_name(n) | コースの名前 |
| Cs_norma(n) | コースのクリアタイム |
| Gm | ゲームモード(0=Gran Prix1=TimeTrial) |
| at0,at1 | タイム算出用 |
| GameFlag | ゲームフラグ |
| PI | 円周率 |
| FieldX,FieldY | フォームの幅,高さ |
| Max_Cs | 登録されているコース数-1 |
| i,j | ループ用 |
| px(n),py(n) | 自機描画用 |
| vx,vz | 自機操作用 |
| その他 | 汎用 |



★OFってすばらしい。ベーマガへの要望や関係のないことまでも書ける。すばらしいコミュニケーションの場です。これからもがんばって！（神奈川県川崎市・**桜井道孝21歳**）……【**編**：コミュニケーションの能力アップにもなるOFです。よろしく！】

## 改造のポイント

RaceMainの中の“r=r+vx＊10”の10を変更することで，旋回性が変わります。大きくすると曲がりやすくなり，小さくすると曲がりにくくなります。

また，同じRaceMain中の“If s > 16 Then s = 16”の2か所の16を変えることで，最高速度が変わります。大きくすると速くなりますが，それだけ曲がるときが難しくなります。

ぜひ，行ってもらいたいのは「コースの追加」でしょう。コースのデータは横20×縦20文字で0～4の数字を使います。0が道，1が芝，2が黄色チェックポイント，3が水色チェックポイント，そして4が赤のチェックポイント（スタート/ゴール地点）となります。

コースを作って名前をつけて保存したら，プログラムのMain文中にある，Cs_name，Cs_file，Cs_normaのそれぞれの最後尾に，コースの名前，コースのファイル名，クリアタイムを追加して，最初のほうにある“Public Const Max_Cs = 2”の2を追加したコースの分だけ増やしてください。クリアタイムは，Time Trialで実際に遊んで出てきた自分のタイムをもとに設定するといいでしょう。

## 作成後記

このプログラムはベーマガ2001年7月号の「Get&Run」を見て，「作ってみようかな」と思いたって組んだプログラムです。また，三角関数を使って自機の移動や表示を行ってみようと思ったのも，作る要因のひとつでした。

さすがに敵車を動かすのは難しいです。そのうえ，リアルタイムに順位を変動させるとなるとなおさらです。

ずっと昔にN-88BASICで画面固定型の見下ろし型レースゲームが，何かの講座であったような気がします。あのプログラムで出てきた車の絵を回転させていくルーチンはすっかり忘れてしまいましたけど。私のメールアドレスはymd3106@jomon.ne.jp，HPアドレスはhttp://www.jomon.ne.jp/~ymd3106/です。感想お待ちしています。HPにも遊びに来てください。

```
End Sub

' レースのメイン画面
Sub RaceMain()
Dim i As Integer, vx As Integer, vz As Integer, p As Long
  If Gm = 0 Then InitStage
  Form1.Cls
  CourseMaking
  r = 0: s = 0: Chk = 0: Lap = 0
  PrintMyChr
  '
  For i = 0 To 2: Msg 250, i * 20 + 10, "LAP" & (i + 1) & ":", 12, 0: Lt(i) = 0: Next i
  Msg 250, 70, "TOTAL TIME", 12, 0
  If Gm = 0 Then
    Msg 250, 110, "CLEAR TIME", 12, 0
    Msg 300, 130, Format(Cs_norma(Cs), "##0.000"), 12, 0
  End If
  tt = 0
  '
```

### 《リスト1》フォームのプログラム・リスト

```
Option Explicit

' フォームを閉じたら終了する
Private Sub Form_Unload(Cancel As Integer)
  End
End Sub

' 時間がきたらラベルを消す
Private Sub Timer1_Timer()
  Form1.LblTime.Visible = False
  Form1.Timer1.Enabled = False
End Sub
```

### 《リスト2》モジュールのプログラム・リスト

```
Option Explicit

Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As Long, ByVal y As Long,
ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal Xsrc As Long, ByVal
Ysrc As Long, ByVal dwRop As Long) As Long
Declare Function GetTickCount Lib "kernel32" () As Long
Declare Function GetAsyncKeyState Lib "user32" (ByVal vKey As Long) As Integer
'
Public x As Integer, y As Integer, r As Integer, s As Integer
Public Chk As Integer, Lap As Integer, Lt(2), tt, Cs As Integer
Public Cs_file As Variant, Cs_name As Variant, Cs_norma As Variant
Public Gm As Integer, at0 As Long, at1 As Long, GameFlag As Integer

Public Const PI = 3.14159   ' 円周率
Public Const FieldX = 380   ' フォームの幅
Public Const FieldY = 264   ' フォームの高さ
Public Const Max_Cs = 2     ' 最大コース数

' 初期設定
Sub Main()
Dim i As Integer
  Form1.Width = Screen.TwipsPerPixelX * FieldX
  Form1.Height = Screen.TwipsPerPixelY * FieldY
  Form1.PicMap.Move 0, 0, 800, 800
  Form1.LblTime.Move 0, 40, 240, 80
  Form1.Caption = "T-Racer for BM (C)Teikoku"
  Form1.Timer1.Enabled = False
  Form1.Show
  ' コースの名前、コースファイル名、Grand Prixでの制限時間
  Cs_name = Array("Trial Course", "Simple Oval", "Radian Lines")
  Cs_file = Array("c1.txt", "c2.txt", "c3.txt")
  Cs_norma = Array(28, 21, 33)
  '
  GameMain
End Sub

' ゲームのメインルーチン
Sub GameMain()
  ' 無限ループ
  While 1
    GamtTitle
    ModeSelect
    Do
      If Gm = 1 Then SelectCourse
      RaceMain
      If Gm = 0 Then Hantei Else Delay 3000: GameFlag = 1
    Loop While (GameFlag = 0)
  Wend
```

★ふははぁ～～っ！この肌触りこそ投稿よぉ～～っ！！（愛知県名古屋市・**ズオウ・リアキ29歳**）……**【影：**おやすみなさ～い。**編集長：**影君に「ふははぁ～～っ」とかいういかにも眠気を誘うような言葉をかけないよう，読者諸君も気をつけてくれたまえ。以上。**】**

```
  Form1.LblTime.Visible = True
  Form1.LblTime.Caption = Cs_name(Cs)
  Delay 3000
  Form1.LblTime.Caption = "Ready..."
  For i = 0 To 2: Beep: Delay 1000: Next i
  Form1.LblTime.Caption = "Go!!": Beep
  Form1.Timer1.Enabled = True
  at0 = GetTickCount(): Chk = 0
  While (Lap < 3)
    vx = (GetAsyncKeyState(37) <> 0) - (GetAsyncKeyState(39) <> 0)
    vz = -(GetAsyncKeyState(32) <> 0)
    If Lap = 3 Then vx = 0: vz = 0
    r = r + vx * 10
    s = s + vz * 2 - 1
    If s < 1 Then s = 0
    If s > 16 Then s = 16
    If r > 360 Then r = r - 360
    If r < 0 Then r = r + 360
    '
    p = Form1.PicMap.Point(x, y)
    If p = QBColor(2) Then s = s / 2
    If s < 1 And vz Then s = 1
    If p = QBColor(14) And Chk = 0 Then Chk = 1: Beep
    If p = QBColor(11) And Chk = 1 Then Chk = 2: Beep
    If p = QBColor(12) And Chk = 2 Then PrintTime: Chk = 0: Beep
    '
    PrintMyChr
    If x < 20 Or x > 780 Or y < 20 Or y > 780 Then CourseOut
    Delay 50
  Wend
  '
  If Lap <> 99 Then Form1.LblTime.Caption = Form1.LblTime.Caption & Chr(13) & "Finish!!"
  While (s > 0)
    PrintMyChr
    s = s - 1
    Delay 50
  Wend
  '
  While (GetAsyncKeyState(32) <> 0): Wend
  If Lap <> 99 Then Msg 300, 90, Format(tt, "##0.000"), 12, 0
  Delay 2000
End Sub

' 自機とその周辺のマップの表示
Sub PrintMyChr()
Dim px(2) As Integer, py(2) As Integer
  x = x + Cos(r / 180 * PI) * s
  y = y + Sin(r / 180 * PI) * s
  BmpCopy Form1.PicMap.hDC, x - 120, y - 120, 240, 240, Form1.hDC, 0, 0
  ' 自機の三角形を描画するため、３つの点の位置を算出
  px(0) = 120 + Cos(r / 180 * PI) * 8: py(0) = 120 + Sin(r / 180 * PI) * 8
  px(1) = 120 + Cos((r + 150) / 180 * PI) * 8: py(1) = 120 + Sin((r + 150) / 180 * PI) * 8
  px(2) = 120 + Cos((r + 210) / 180 * PI) * 8: py(2) = 120 + Sin((r + 210) / 180 * PI) * 8
  ' ３つの点を直線で結び、三角形を表示
  Form1.Line (px(0), py(0))-(px(1), py(1)), vbBlue
  Form1.Line (px(1), py(1))-(px(2), py(2)), vbBlue
  Form1.Line (px(2), py(2))-(px(0), py(0)), vbBlue
  '
  Form1.Refresh
End Sub

' ラップタイム算出・表示
Sub PrintTime()
Dim t
  at1 = GetTickCount
  t = (at1 - at0) / 1000
  Lt(Lap) = t: tt = tt + t
  Form1.LblTime.Caption = Format(t, "##0.000")
  Msg 300, Lap * 20 + 10, Format(t, "##0.000"), 12, 0
  Lap = Lap + 1
  Form1.LblTime.Visible = True
  at0 = at1
  Form1.Timer1.Enabled = True
End Sub

' コースアウトの判定
Sub CourseOut()
  Form1.LblTime.Caption = "Course Out!!"
  Form1.LblTime.Visible = True
  Form1.Timer1.Enabled = True
  Lap = 99
End Sub

' Grand Prixモードでのクリア判定
Sub Hantei()
  Form1.LblTime.Caption = ""
  Form1.LblTime.Visible = True
  If Lap <> 99 And tt <= Cs_norma(Cs) Then
    Form1.LblTime.Caption = "Stage Clear!!"
    Delay 3000
    If Cs = Max_Cs Then
      Form1.LblTime.Caption = "Congratulations!"
      Delay 3000
      GameFlag = 1
    Else
      Cs = Cs + 1
      GameFlag = 0
    End If
  Else
    Form1.LblTime.Caption = "Game Over..."
    Delay 3000
    GameFlag = 1
  End If
  Form1.LblTime.Visible = False
End Sub

' Time Trialモードでのコース選択
Sub SelectCourse()
Dim f As Integer, vx As Integer
  Form1.Cls
  While (GetAsyncKeyState(32) <> 0): Wend
  Msg 80, 20, "Course Select", 24, 9
  Cs = 0: f = 0
  Msg 20, 100, "<", 16, 0: Msg 320, 100, ">", 16, 0
  Msg 60, 180, "Push Space Key To Start", 16, 0
  Msg 160 - (Len(Cs_name(Cs)) / 2) * 8, 100, Cs_name(Cs), 16, 0
  Form1.Refresh
  While (f = 0)
    vx = (GetAsyncKeyState(37) <> 0) - (GetAsyncKeyState(39) <> 0)
    Cs = Cs + vx
    If Cs < 0 Then Cs = Max_Cs
    If Cs > Max_Cs Then Cs = 0
    If vx <> 0 Then
      Form1.Line (40, 100)-(300, 120), Form1.BackColor, BF
      Msg 160 - (Len(Cs_name(Cs)) / 2) * 8, 100, Cs_name(Cs), 16, 0
      Form1.Refresh
    End If
    If (GetAsyncKeyState(32) <> 0) Then f = 1
    Delay 100
  Wend
End Sub
```

★マーク模試で初めてBASICに挑戦。もちろん満点。ベーマガ万歳！！（愛媛県東宇和郡・**毛利祐允17歳**）……【**編**：いよいよセンター試験も近づいてきましたね。がんばってくださいね。次号では恒例の「センター試験対策BASIC予想問題」を掲載します。】

```
' タイトル画面
Sub GamtTitle()
Dim f As Integer
  Form1.Cls
  While (GetAsyncKeyState(32) ◇ 0): Wend
  Msg 20, 20, "T-Racer for BM", 36, 9
  Msg 60, 110, "Push Space Key To Start", 16, 0
  Msg 50, 200, "(C)2001 Teikoku All Rights Reserved.", 12, 0
  '
  f = 0
  While (f = 0)
    If (GetAsyncKeyState(32) ◇ 0) Then f = 1
    DoEvents
  Wend
  '
End Sub

' モードセレクト画面
Sub ModeSelect()
Dim f As Integer, vx As Integer, mode As Variant
  Form1.Cls
  mode = Array("Grand Prix", "Time Attack")
  While (GetAsyncKeyState(32) ◇ 0): Wend
  Msg 100, 20, "Mode Select", 24, 9
  Gm = 0: f = 0
  Msg 20, 100, "<", 16, 0: Msg 320, 100, ">", 16, 0
  Msg 60, 180, "Push Space Key To Start", 16, 0
  Msg 160 - (Len(mode(Gm)) / 2) * 8, 100, mode(Gm), 16, 0
  Form1.Refresh
  While (f = 0)
    vx = (GetAsyncKeyState(37) ◇ 0) - (GetAsyncKeyState(39) ◇ 0)
    If vx ◇ 0 Then
      Gm = 1 - Gm
      Form1.Line (40, 100)-(300, 120), Form1.BackColor, BF
      Msg 160 - (Len(mode(Gm)) / 2) * 8, 100, mode(Gm), 16, 0
      Form1.Refresh
    End If
    If (GetAsyncKeyState(32) ◇ 0) Then f = 1
    Delay 100
  Wend
  GameFlag = 0
  If Gm = 0 Then Cs = 0
End Sub

' Grand Prixモードでのステージ開始画面
Sub InitStage()
  Form1.Cls
  Msg 120, 10, "Stage " & (Cs + 1), 24, 9
  Msg 160 - (Len(Cs_name(Cs)) / 2) * 10, 50, Cs_name(Cs), 20, 0
  Msg 60, 100, "Clear Time:" & Format(Cs_norma(Cs), "#0.000"), 24, 0
  Msg 100, 160, "Good Luck!!", 24, 0
  Delay 3000
End Sub

' コースの作成
Sub CourseMaking()
Dim i As Integer, j As Integer, fnum As Integer, d As String, c As Long
  Form1.PicMap.Cls
  fnum = FreeFile
  ' 指定されたファイルからデータを読み込む
  Open MyForm & Cs_file(Cs) For Input As #fnum
  For j = 0 To 19
    Input #fnum, d
    For i = 0 To 19
      c = Val(Mid$(d, i + 1, 1))
      ' 赤のチェックポイントの部分がスタート地点になる
      If c = 4 Then x = i * 40 + 20: y = j * 40 + 20
      c = Val("&H" & Mid$("2febc", c + 1, 1))
      Form1.PicMap.Line (i * 40, j * 40)-(i * 40 + 39, j * 40 + 39), QBColor(c), BF
    Next i
  Next j
  Close #fnum
  Form1.PicMap.Refresh
End Sub

' フォーム上にメッセージを表示
Sub Msg(ByVal x As Long, ByVal y As Long, ByVal n As String, ByVal s As Integer, ByVal c As
Integer)
  Form1.CurrentX = x: Form1.CurrentY = y: Form1.FontSize = s: Form1.ForeColor = QBColor(c)
  Form1.Print n
End Sub

' 画像転送
Function BmpCopy(sp, sx, sy, w, h, gp, gx, gy)
Dim exec
  exec = BitBlt(gp, gx, gy, w, h, sp, sx, sy, &HCC0020)
End Function

' 時間待ち
Function Delay(ByVal t As Long)
Dim At_0 As Long, At_1 As Long
  At_0 = GetTickCount
  At_1 = At_0
  While ((At_1 - At_0) < t)
    At_1 = GetTickCount
    DoEvents
  Wend
End Function

' 実行時のフォルダ名取得
Function MyForm()
  If Right$(App.Path, 1) ◇ "¥" Then
    MyForm = App.Path & "¥"
  Else
    MyForm = App.Path
  End If
End Function
```

**《リスト３》c1.txtのリスト**

```
00000000000000000000
00000000000000000000
00011141111111111000
00111141111111111100
00110000000000001100
00110000000000001100
00111000000000001100
00111100000011111100
00001100000011111000
00001100000110000000
00011100000220000000
00111000000111000000
00110000000011100000
00110000000001100000
00111000000011100000
00011100001111100000
00001113111111000000
00000113111000000000
00000000000000000000
00000000000000000000
```

**《リスト４》c2.txtのリスト**

```
00000000000000000000
00000000000000000000
00001111141111110000
00011111141111111000
00011000000000011000
00111000000000011100
00111000000000011100
00111000000000011100
00333000000000011100
00111000000000011100
00111000000000011100
00111000000000111000
00011000000001110000
00011111121111100000
00001111121111100000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
```

**《リスト５》c3.txtのリスト**

```
00000000000000000000
00000000000000000000
00001111141111111100
00001111141111111100
00111100000000001100
00111100000000001100
00110000000000001100
00330000000000001100
00111111111111001100
00111111111111001100
00000000000011001100
00000000000011001100
00111111111111001100
00111111111111001100
00220000000000001100
00110000000000001100
00111111111111111100
00111111111111111100
00000000000000000000
00000000000000000000
```



★友人曰く。ベーマガ買わなくてもいいから投稿しろ，だそうです。いいのか？（東京都練馬区・どらねく15歳）……【影：いいわけないだろう。「ベーマガ買って，投稿もして」。これが，ベーマガの読みかたの基本。Dr.D：影君もたまにはいいことを言うのう。】

（超大作プログラム）　リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net/)からダウンロードしてお楽しみください。

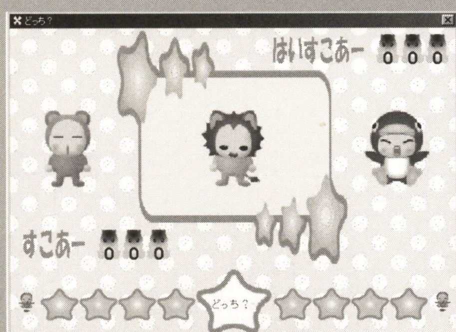
# Windows(Visual Basic V6.0+DirectAnimation)用



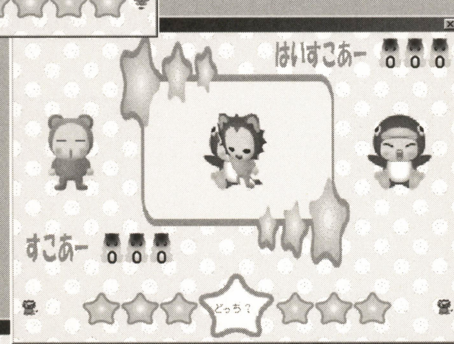
変化後の画像はどっちだ？

# どっち？

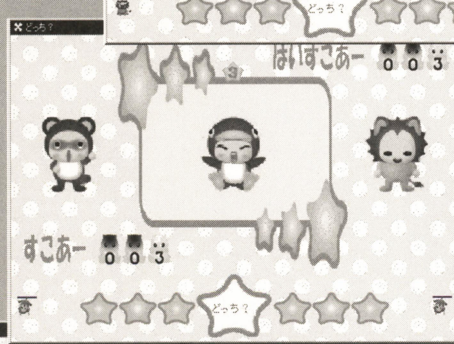
橘　一馬（29歳）




《写真１》中央のキャラクターの変化に注目




《写真２》トランスフォームが適用され，別のキャラに




《写真３》左右どちらのキャラかを素早く答える




《写真４》数多くのトランスフォームが用意されているゾ

## 遊びかた

真ん中の絵が変わっていきます。左の絵だと思ったら左クリック，右の絵だと思ったら右クリックをしてください。

問題が進むごとに，星が少しずつ減っていきます。星がなくなるとゲーム・オーバー。正解すると少し星が増えます。間違うと星が減ります。

## プログラム

メインの絵が変わるところは，ベーマガ2000年2月号のDirectX講座のプログラムそのものです。

## CHECKER FLAG

**Dr.D**：画像ファイルが多かったため，超大作での掲載とさせてもらった。この作品は,VBからDirectAnimationを利用したプログラム。画面構成は見栄えがするが，プログラムの肝となる部分は，DirectAnimationを利用していることもあり，シンプルな構成じゃ。

**編**：橘さん作品の作成面での特徴は，他の人が作成した素材やサンプルをうまく利用している点ですね。投稿者の中には，すべて自分で作る，という人もいますが，使えるものはどんどん使う，という柔軟な姿勢も作品を作っていくうえでは必要かもしれません。

**つぐ美**：ただ，他の人が作成した素材を使用した作品を投稿するときは，著作権や配布条件に注意してくださいね。ベーマガで採用された場合は，原稿料が発生するため，「私的利用のみ許可」といった条件のときは注意が必要です。



★どこにもつながっていない。PS2しか持っていない。そんな立場から言うと「試聴室」等は楽しめないのでCDまたはDVDの付録がたまに付いたらいいなぁ（…なんて言ってみただけ）。パソコンほしい。（愛知県名古屋市・わいち301 27歳）……【影：ははーっ。】

「DirectAnimationってこんなこともできるんだぁー」と毎回楽しみにしてます。

質問なんですが，DirectAnimationが実行されている最中に乱数を発生させると，たまにエラーが発生します。そういうもんなのでしょうか？

後のプログラムはおまけみたいなものです。なんかいろいろとくっつけちゃいました。

最初は，WAVファイルでBGMを作ったのですが，容量が大きくてMP3にしました。ループするとき一瞬，止まるのが気になります。DirectXでMP3は演奏できないのでしょうか？DirectX8ではできるみたいなのですが，自分がプログラミングしているのはDirectX7です。

## 協力してくれた人

「イラそよ」のフリー素材を利用させていただきました。ありがとうございました。

### Dr.Dのワンポイント解説

**Dr.D**：この部分リストは，説明文にもあるとおり2000年2月号のDirectX講座のリストを元に作成されておる。特徴としては，トランスフォームの種類を“dxtArr”という配列にあらかじめ設定しておき，乱数の値を添字にすることで，実行するたびにいろんなトランスフォームが実行されるわけじゃ。

また，トランスフォームのイメージを作成する前に，正解となる画像の判断を行うのも忘れてはならん部分じゃ。これも乱数によって左右どちらかの絵を選び，その絵を変化後の表示画像として設定しておる。

**《リスト1》トランスフォーム適用部分のリスト**

```
●Pick Up LIST
'左右の絵を表示
    ImgLNo = Int(Rnd(1) * CHAR_MAX)
    ImgRNo = Int(Rnd(1) * CHAR_MAX)
    imgL = LoadPicture(App.Path & "¥Char" & ImgLNo & ".gif")
    imgR = LoadPicture(App.Path & "¥Char" & ImgRNo & ".gif")

'空のイメージをセット
    DAControl.Image = m.EmptyImage

'旧トランスフォームの解放
    Set dxt = Nothing

'プログラムＩＤの作成
    TransformNo = Int(Rnd(1) * 22)
    progID = "DXImageTransform.MetaCreations." & dxtArr(TransformNo)(1)

'入力イメージ
    On Error GoTo 0
    If Int(Rnd(1) * 2) = 0 Then
        ImgBNo = ImgLNo
    Else
        ImgBNo = ImgRNo
    End If
    Set imgA = m.ImportImage(App.Path & "¥Char" & ImgANo & ".gif")
    Set imgB = m.ImportImage(App.Path & "¥Char" & ImgBNo & ".gif")
    inputs = Array(imgA, imgB)
```




by 吏久

評…「オペラ座の怪人」ね。私も見てみたい（つぐ美）



by こがらす丸様江

評…何してるんだろう？気になって夜も眠れない（影）




by ていこく

評…出たな！～計画！有名になりましたか？（編）




by ぶんちょ

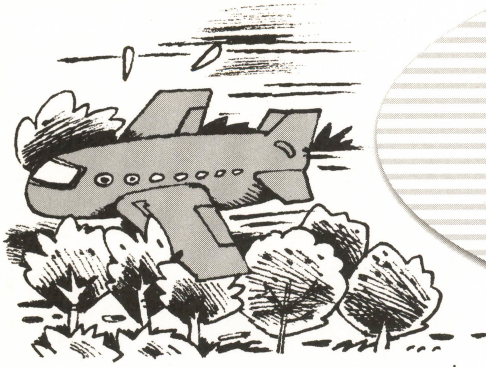
評…これ、私だそうです。ありがとうね（つぐ美）

★最近OFの締め切りが守れません。なので，2号あとのOFを目指すことにしました。（岩手県釜石市・鉄人坂井23歳）……【影：私の力で，2号あとなどと言わずに，12月号に載せてやったぞ。編：だめですよ。彼の方針がまたくずれたってことでしょ。影：ぐはは。】

# Windows(VBScript + DirectX)用

ブラウザで遊べるフライトシミュレータ

# Touch Down（タッチ ダウン）



金井　義男（39歳）

## あらすじ

免許を取得し，真っ先に向かった彼女の家の上空で，着陸寸前ナビが故障した。マニュアル操作で着陸をしなければならなくなった。でも，免許取り立ての僕は着陸がにがて，手に汗をにぎりながら着陸開始。

## 内容

飛行着陸シミュレーション的なゲームです。ベーマガ2000年8月号の,Height Fieldトランスフォームを使ったサンプルをベースに作成しました。

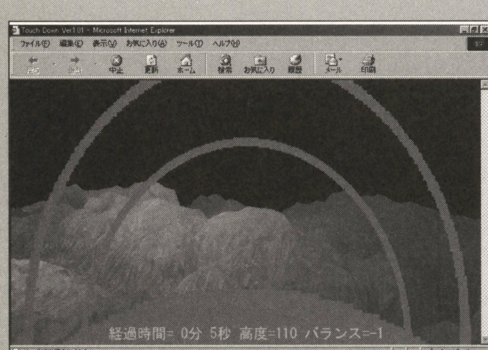
## 使いかた・遊びかた

飛行シミュレーションを楽しんで，着陸に挑戦します。↑，↓キーで高度を変更し，←，→キーでバランスを取りながら(0にする)，↑，↓キーを押し続けるとスピードが増減します。

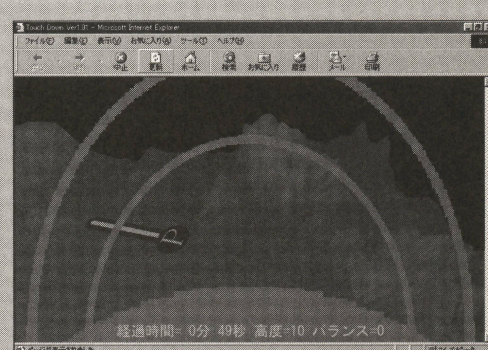
着陸は，最低高度(10)まで下げたうえで，滑走路のPマークが目前に来たところで，Ctrlキーで着陸を開始します。Enterキーでゲームを開始。Escキーでタイトル画面に戻ります。着陸に失敗すると，「WARNING SafeGuard」メッセージが表示され，ペナルティが科せられます。高度が最高点に達すると降下速度がUPして，より短期間で着陸できるようになります。

## プログラムの入力法と注意点

メモ帳を使って入力，改造ができますが，デバッガとテキストエディタ機能のあるホームページ作成ツールを使用することをおすすめします。



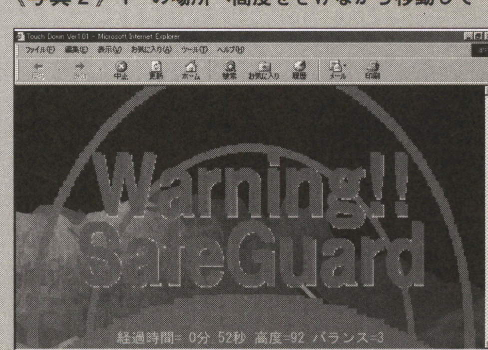

《写真１》3Dフライトをブラウザで体験？




《写真２》“P”の場所へ高度をさげながら移動して…




《写真３》Ctrlキーで着陸。うまく着陸できるか？




《写真４》着陸失敗だと，この表示。もう一度チャレンジだ

## 改造のアドバイスと苦労したところ

単純なゲームなので，色々な改造できそうですね。クッキーを使って着陸最短時間を記録できないかと思ったんですが，VBScriptからJavaScriptの変数を受け渡しできますかね？

カメラアングルの配置は，ふだん気を使わなかったのでむずかしいですね（というわけで，サンプルどおりの地形とアングルをそのまま使用しているのです)。ホームページでも紹介しています。お暇な方は http://www.d7.dion.ne.jp/~kanaisunへ



★受験生になると雑誌に投稿したくてたまらなくなる法則。きっちりはまっちゃいました。ハハハハ…。（大阪府堺市・三匹目のryo17歳）……【つぐ美：気分転換に，ベーマガへの投稿もよろしいかと思いますわよ。OF，イラストなんでもお待ちしてますわよ。】

## CHECKER FLAG

**つぐ美**：DirectAnimationということもあって，細かなゲーム設定(地形との当たり判定etc)は省略された形ですが3D表示のため見栄えがGoodです。

**影**：ただ，このリストのままの操作だと，↑で高度を上げるという，一般的なフライトゲームの設定と逆のため操作しにくい。この部分は，各自の好みに合わせてキー入力部分のCaseの値を書き換えよう。

**Dr.D**：基本はサンプルを使用しているとはいえ，なかなかの出来映えじゃろう。ただ，ゲームとしては，高度を10にして，着陸地点まで移動したらCtrlを押せばクリアできてしまうため，おもしろ味に欠ける。ゲーム性の面で，もう一工夫ほしかったな。

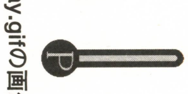
**編**：実行に必要な画像ファイルについてですが，"ending.jpg"ファイルは写真データで，作成が困難と思われるため，誌面への掲載はしませんでした。この"ending.jpg"は各自で用意してください。ちなみにオリジナル画像のサイズは250×160なので，これに合わせるのが良いでしょう。



pit.gifの画像



runwayの画像



way.gifの画像




warning.gifの画像




title.gifの画像



picture.jpgの画像

### 『Touch Down』のプログラム・リスト

```
<HTML>
<TITLE>Touch Down Ver1.01</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0", BGCOLOR = "#0">
<SCRIPT LANGUAGE = "VBScript">
Option Explicit
'-----------------------------------------------------------
'初期化処理
'-----------------------------------------------------------
  Dim m          'As DAStatics
  Dim mx         'As DANumber    飛行左右範囲
  Dim mz         'As DANumber    飛行高度範囲
  Dim ms         'As DANumber    飛行速度
  Dim zm         'As DANumber    着地用拡大率
  Dim title      'As DAImage     タイトル画像読み込みイメージ
  Dim titleimg   'As DAImage     タイトルイメージ
  Dim warning    'As DAImage     着陸警告画像読み込みイメージ
  Dim dispimg    'As DAImage     パラメータ表示 イメージ
  Dim cw         'As DANumber    クライアント ウィンドウの幅
  Dim ch         'As DANumber    クライアント ウィンドウの高さ
  Dim pitimg     'As DAImage     コックピット表示 イメージ
  Dim pitrot     'As DANumber    ピット回転角度
  Dim pitang     'As DANumber    ピット上下回転角度
  Dim runway     'As DAImage     エンディング着陸滑走路画像読み込みイメージ
  Dim runimg     'As DAImage     着陸滑走路表示 イメージ
  Dim endimg     'As DAImage     終了イメージ
  Dim ending     'As DAImage     終了イメージ格納変数
  Dim geoimg     'As DAImage     レンダリング イメージ
  Dim pw         'As DANumber    ピット ウィンドウの幅
  Dim ph         'As DANumber    ピット ウィンドウの高さ
  Dim snd1       'As DASound     BGMサウンド
  Dim snd2       'As DASound     着陸音
  Dim newsnd1    'As DASound     ゲーム中BGMサウンド
  Dim newsnd2    'As DASound     着陸音
  Dim newsnd3    'As DASound     エンディングBGMサウンド
  Dim evtcnt     'As DANumber    ゲームスタートからのカウント１００で１秒
  Dim downval    'As Long        高度下げ幅値
  Dim flag       'As Boolean     着陸成功判定（成功：True、未：False）
  Dim nowcnt     'As Long        着陸ボタンDown時のカウント数一時保存変数
  Const PI       = 3.14159265359  '円周率
Sub Window_onLoad()
  '変数の宣言
  Dim finalimg   'As DAGeometry     最終 イメージ
  Dim evt        'As DAEvent        タイマー用イベント
  Dim none       'As DASound        イベントハンドリング
  Dim snd        'As DASound        効果音

  'DirectAnimationの初期化（メートル座標系）
  Set m = DAControl.MeterLibrary
  flag = True    'スタート前のフラグ
  '------------------------------
  'コックピットイメージの作成
  '------------------------------
  Call PitMake   'コックピットイメージの作成
  Call Disppara  '画面表示パラメータイメージの作成
  Call PlnMake   '着陸地全景イメージの作成
  '------------------------------
  'タイマーイベント
  '------------------------------
  Set evt = m.Timer(0.01)
  Set evt = evt.ScriptCallback("NextTime", "VBScript")
  Set none = CreateObject("DirectAnimation.DASound")
  none.Init m.Until(m.Silence, evt, none)
  '------------------------------
  '効果音をビヘイビアでの合成(ミキシング)
  '------------------------------
  Set newsnd1 = m.ImportSound("sound.mp3").Sound.Loop    '無限リピート再生するDASoundオブジェク
ト
  Set newsnd2 = m.ImportSound("takeon.mp3").Sound        '着陸音DASoundオブジェクト
  Set newsnd3 = m.ImportSound("ending.mp3").Sound.Loop   '無限リピート再生するDASoundオブジェク
ト
  Set snd1 = m.ModifiableBehavior(m.Silence)    'サウンドを無限リピート再生するDASoundオブジェ
クト初期値                                       '無限を意味するDASoundオブジェ
クト
  Set snd2 = m.ModifiableBehavior(m.Silence)    '着陸音DASoundオブジェクト初期値
  Set snd = m.Mix(snd1, snd2)
  '------------------------------
  'トータルイメージの作成
  '------------------------------
  'Array関数は、配列形式のバリアント型変数を返します。（配列は、絵の重なる最上位から順に）
  Set finalimg = m.OverlayArray(Array(dispimg, ending, pitimg, geoimg))
  'コントロールにオブジェクトをセットしてスタート
  DAControl.Image = finalimg
  DAControl.Sound = m.Mix(snd, none)
  DAControl.Start
  Call Window_onResize
End Sub

'-----------------------------------------------------------
'ウィンドウのサイズ変更時の処理
'-----------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
  'クライアントの幅と高さのビヘイビアを変更(m.Pixel = 0.000263)
  cw.SwitchToNumber Document.Body.ClientWidth * m.Pixel.Extract / 2.1
  ch.SwitchToNumber Document.Body.ClientHeight * m.Pixel.Extract / 2.1
  'コックピットの幅と高さのビヘイビアを変更
```

★今晩は。はじめてこの雑誌を買いました。父からもらったおさがりのパソコン（でも一応Windows95対応です）ですが，いろいろと学んでいきたいです。ゲーム以外でもいろいろ使いたいと思います。（愛知県知多郡・TAプロトタイプ26歳）……【影：よろしく！】

```
  pw.SwitchToNumber Document.Body.ClientWidth / 400
  ph.SwitchToNumber Document.Body.ClientHeight / 180
  DAControl.focus
End Sub
'------------------------------------------------------------
'NextTime : タイマーカウンター
'------------------------------------------------------------
Private Sub NextTime()
  ms.SwitchToNumber (ms.Extract - 0.001)
    If ms.Extract < -2 * PI Then ms.SwitchToNumber 0
  If flag = False Then
    evtcnt.SwitchToNumber evtcnt.Extract + 1
      If evtcnt.Extract Mod (60 * 100) = 0 Then mx.SwitchToNumber (mx.Extract + 0.01)
      If evtcnt.Extract > nowcnt + 500 Then titleimg.SwitchTo m.EmptyImage
  End if
End Sub
'------------------------------------------------------------
'画面表示パラメータイメージの作成
'------------------------------------------------------------
Private Sub Disppara()
  Dim ds      'As DAString  DirectAnimation文字列
  Dim num     'As DANumber
  Dim fnt        'As DAFontStyle  フォント
  '------------------------------
  'パラメータをビヘイビアとして定義
  '------------------------------
  Set cw = m.ModifiableNumber(0)         'DAComtrol表示幅をビヘイビアとして定義
  Set ch = m.ModifiableNumber(0)         'DAComtrol表示高さをビヘイビアとして定義
  Set mx = m.ModifiableNumber(1)         'カメラ位置X値をビヘイビアとして定義
  Set mz = m.ModifiableNumber(0)         'カメラ位置Z値をビヘイビアとして定義
  Set ms = m.ModifiableNumber(0)         '着陸地全景回転角度をビヘイビアとして定義
  Set pitang = m.ModifiableNumber(0)     'カメラアングルをビヘイビアとして定義
  Set evtcnt = m.ModifiableNumber(0)     '経過時間カウンターをビヘイビアとして定義
  '-----------------------------------
  '各種パラメータ表示イメージの作成
  '-----------------------------------
  Set num = m.DANumber(100)
  'DANumber型の値を引数で指定された小数点桁数を持つDAString型に変換し、文字列の結合
  Set ds = m.DAString("経過時間= ")   'cm単位で表示する
  Set ds = m.ConcatString(ds, m.Floor(m.Div(m.Div(evtcnt, num), m.DANumber(60))).ToString(0))
  Set ds = m.ConcatString(ds, m.DAString("分 "))
  Set ds = m.ConcatString(ds, m.Mod(m.Div(evtcnt, num), m.DANumber(60)).ToString(0))
  Set ds = m.ConcatString(ds, m.DAString("秒 高度="))
  Set ds = m.ConcatString(ds, m.Add(m.Mul(mz, num), m.DANumber(110)).ToString(0))
  Set ds = m.ConcatString(ds, m.DAString(" バランス="))
  Set ds = m.ConcatString(ds, m.Sub(m.Mul(mx, num), m.DANumber(90)).ToString(0))
  'Set ds = m.ConcatString(ds, m.DAString(" 位置="))
  'Set ds = m.ConcatString(ds, ms.ToString(2))  'イメージの回転表示角度のラジアン値
  'フォントの定義
  Set fnt = m.DefaultFont.Family("ＭＳ ゴシック").Size(18).Color(m.Green)
  'DAStringからDAImageを作成
  Set dispimg = m.StringImageAnim(ds, fnt)
  '常に画面下側で表示（ビヘイビアを使用）
  Set dispimg = dispimg.Transform(m.Translate2Anim(m.DANumber(0), m.Mul(ch, m.DANumber(-1))))
End Sub
'------------------------------------------------------------
'コックピットイメージの作成
'------------------------------------------------------------
Private Sub PitMake()
  Dim trn      'As DANumber
  Dim titleop  'As DAImage

  Set pw = m.ModifiableNumber(0)                 '表示サイズをビヘイビアとして定義
  Set ph = m.ModifiableNumber(0)
  Set pitrot = m.ModifiableNumber(0)             '回転角度をビヘイビアとして定義
  Set title = m.ImportImage("title.gif")         'タイトル画像データの読み込み
  Set title = title.Transform(m.Translate2(0, 0.002))  '上下位置合わせ
  Set titleimg = m.ModifiableBehavior(title)     'タイトル画像をビヘイビアとして定義
  Set pitimg = m.ImportImage("pit.gif")          'コックピット画像データの読み込み
  Set runway = m.ImportImage("runway.gif")       '着陸滑走路面画像データの読み込み
  'DANumberオブジェクトを定義
  ' [-0.1が3秒間（継続）] と [-0.1～0が5秒間（継続）] を合成
  Set trn = m.Sequence(m.Interpolate(-0.1, -0.1, 3), m.Interpolate(-0.1, 0, 5))
  Set runway = runway.Transform(m.Translate2Anim(m.DANumber(0), trn))
```

```
  Set runway = runway.Transform(m.Scale2UniformAnim(m.Interpolate(0.5, 1, 10)))
  Set runimg = m.ModifiableBehavior(m.EmptyImage)'滑走路面画像をビヘイビアとして定義
  Set pitimg = m.OverlayArray(Array(titleimg, pitimg, runimg))  'イメージを合成
  Set pitimg = pitimg.Transform(m.Scale2Anim(pw, ph))   'イメージの縮小拡大変換
  Set pitimg = pitimg.Transform(m.Rotate2Anim(pitrot))  'イメージの回転変換
  Set warning = m.ImportImage("warning.gif")  '警告メッセージ画像データの読み込み
  Set endimg = m.ImportImage("ending.jpg")   '終了イメージ画像データの読み込み
  Set trn = m.Sequence(m.Interpolate(0, 1, 15), m.Interpolate(1, 0, 10))
  Set titleop = title.OpacityAnim(m.Interpolate(0, 1, 20))  'タイトルイメージ透過変換
  Set titleop = titleop.Transform(m.Scale2Anim(pw, ph))     'イメージの縮小拡大変換
  Set endimg = endimg.Transform(m.Scale2UniformAnim(trn))   'イメージの縮小拡大変換
  Set endimg = m.Overlay(endimg, titleop)           '終了イメージを合成
  Set ending = m.ModifiableBehavior(m.EmptyImage)   '終了イメージをビヘイビアとして定義
End Sub
'------------------------------------------------------------
'着陸地イメージの作成
'------------------------------------------------------------
Private Sub PlnMake()
  Dim img         'As DAImage
  Dim dxt         'As DXTMSFT3Lib.HeightField
  Dim res         'As IDADXTransformResult
  Dim geo         'As DAGeometry
  Dim wayimg      'As DAGeometry
  Dim light       'As DAGeometry
  Dim camera      'As DACamera
  Dim tran        'As DATransform3
  Dim progress    'As DANumber
  Dim inputs      'As Variant
  Set zm = m.ModifiableNumber(1)
  '-----------------------------------
  '着陸地全景イメージの作成
  '-----------------------------------
  'DirectX Transformの作成（HeightFieldトランスフォーム）
  Set dxt = CreateObject("DX3DTransform.Microsoft.HeightField")
  'DirectX Transformプロパティの設定
  dxt.Depth = 1
  dxt.Width = 4
  dxt.Height = 4
  '入力オブジェクトの定義（イメージ）
  Set img = m.ImportImage("picture.jpg")
  inputs = Array(img)
  'プログレス値の設定
  Set progress = m.DANumber(1)
  'DirectX Transform の適用
  Set res = m.ApplyDXTransform(dxt, inputs, progress)
  'トランスフォーム適用後のジオメトリを取得
  Set geo = res.OutputBvr
  '-----------------------------------
  '滑走路イメージの作成
  '-----------------------------------
  'DirectX Transform オブジェクトの作成（Shapes）
  Set dxt = CreateObject("DX3DTransform.Microsoft.Shapes")
  'DirectX Transform の適用
  Set res = m.ApplyDXTransform(dxt, Null, Null)
  res.PutBvrAsProperty "Shape", m.DAString("Sheet")
  Set wayimg = res.OutputBvr
  'テクスチャイメージのロード
  Set img = m.ImportImage("way.gif")
  'イメージの上下反転（マッピング時ポリゴンの頂点順序で、内側から貼り付けされるため））
  Set img = img.Transform(m.Scale2(1, -1))
  '物体にイメージを貼り付ける(マッピング)
  Set wayimg = wayimg.Texture(img.MapToUnitSquare)
  Set wayimg = wayimg.Transform(m.Scale3(0.1, 0.3, 1))  'イメージの縮小変換
  '移動配置の定義
  Set wayimg = wayimg.Transform(m.Translate3(1.25, 0, 0.5))
  '回転配置の定義
  Set wayimg = wayimg.Transform(m.Rotate3(m.ZVector3, -0.35))
  '着陸地全景イメージとの合成
  Set geo = m.UnionGeometry(geo, wayimg)
  '-----------------------------------
  '光源の定義（ポイント・ライト）
  '-----------------------------------
  Set light = m.PointLight.Transform(m.Translate3(0, 0, 2))
  Set geo = m.UnionGeometry(geo, light)
```

★ASIMOとか一度，生で見たいと思うのですが，時間とお金がありません。岐阜にももっとそんなイベントがあったらいいなぁ，と思います。（岐阜県本巣郡・蜂（bee!!）15歳）……**【編：ASIMOですか。いろんな場所に引っ張りだこのようです。岐阜にも近々？】**

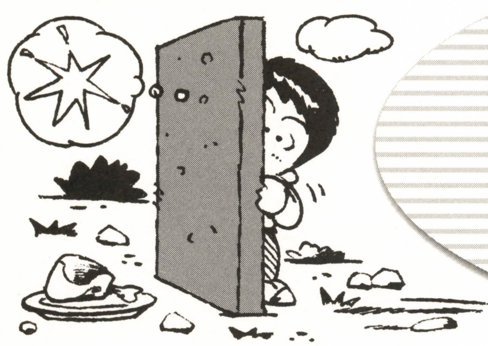
```
    'ジオメトリの回転と移動
    Set geo = geo.Transform(m.Rotate3Anim(m.ZVector3, ms))
    Set geo = geo.Transform(m.Rotate3Degrees(m.XVector3, -80))
    Set geo = geo.Transform(m.Translate3(0, -0.4, 0))
    'カメラの定義 (パースペクティブ・カメラ)
    Set camera = m.PerspectiveCamera(1, 0.9999)
    Set camera = camera.Transform(m.Scale3Anim(m.DANumber(10), m.DANumber(10), zm))
    'カメラのアングルの設定
    Set camera = camera.Transform(m.Rotate3Anim(m.XVector3, pitang))
    'カメラの位置(制限値、mx:0.4〜1.4,mz:-1〜1)
    Set tran = m.Translate3Anim(mx, m.DANumber(0.5), mz)
    Set camera = camera.Transform(tran)
    Set geoimg = geo.Render(camera)
  End Sub
  '------------------------------------------------------------
  '着陸成功か否かの判定
  '------------------------------------------------------------
  Private Sub Check()
    If mx.Extract > 0.89 And mx.Extract < 0.91 And mz.Extract < -0.95 And _
       ms.Extract < -4.9 And ms.Extract > -5.2 Then
      flag = True
      snd1.SwitchTo newsnd3
      runimg.SwitchTo runway
      ending.SwitchTo endimg
    Else
      '着陸失敗時のペナルティ
      titleimg.SwitchTo warning
      downval = 0.03
      mz.SwitchToNumber 0
      ms.SwitchToNumber (ms.Extract - 0.5)
      mx.SwitchToNumber (mx.Extract + 0.03)
        If mx.Extract > 1.4 Then mx.SwitchToNumber 1.4
      nowcnt = evtcnt.Extract
    End if
  End Sub
  '--------------------------------------------------
  'ResetParameters : 設定値の初期化
  '--------------------------------------------------
  Function ResetParameters()
    evtcnt.SwitchToNumber 0
    ms.SwitchToNumber 0
    mz.SwitchToNumber 0
    zm.SwitchToNumber 1
    runimg.SwitchTo m.EmptyImage
    ending.SwitchTo m.EmptyImage
  End Function
  '--------------------------------------------------
  'DAControl_KeyDown : キー操作に対応する処理
  '--------------------------------------------------
  Private Sub DAControl_KeyDown(ByVal KeyCode, ByVal KeyData)

    Select Case KeyCode
      Case 13  '[Enter]:ゲームスタート
        '各種設定の初期化
        Call ResetParameters
        Randomize   ' 乱数発生ルーチンを初期化します。
        mx.SwitchToNumber Int((20 * Rnd) + 81) / 100      '0.81から1.10 までの乱数
        downval = 0.03    '↓KeyDown時の減少値
        snd1.SwitchTo newsnd1               'BGMサウンドイメージの切り替え
        titleimg.SwitchTo m.EmptyImage      'タイトル画像イメージの切り替え
        flag = False                        '着陸判定フラグ
      Case 17  '[Ctrl]:着陸開始
        If KeyData < 100000000 And KeyData > 0 And flag = False Then
          ms.SwitchToNumber (ms.Extract + 0.01)
          snd2.SwitchTo newsnd2             '着陸音イメージの切り替え (適用)
          zm.SwitchToNumber 0.1             'カメラをズームアップし臨場感をUPする
          Call Check                        '着陸成功か否かチェックする
        ElseIf KeyData < 0 And flag = False Then
          zm.SwitchToNumber 1               'ボタンUP時カメラのズームアップを解除する
        End If
      Case 27  '[ESC]:タイトルの表示
        flag = True
        titleimg.SwitchTo title             'タイトル画像イメージの切り替え
        snd1.SwitchTo m.Silence             'BGMサウンドイメージの切り替え
        Call ResetParameters                'パラメータの初期化
      Case 38  '[↑]:高度を上げる (スピードアップ)
        If flag = False Then
          ms.SwitchToNumber (ms.Extract - 0.02)
          mz.SwitchToNumber (mz.Extract + 0.05)
            If KeyData > 0 Then
              If mz.Extract < 0 Then
                pitang.SwitchToNumber 0.1
                ms.SwitchToNumber (ms.Extract - 0.01)
              ElseIf mz.Extract < 1 Then
                pitang.SwitchToNumber 0.4
                ms.SwitchToNumber (ms.Extract - 0.01)
              Else        '高度が最高に達したら↓keyDown時の降下値をUPする
                mz.SwitchToNumber 1
                pitang.SwitchToNumber 0
                downval = 0.05
              End if
            Else          'ボタンUP時カメラのアングルを変更する
              pitang.SwitchToNumber 0
            End If
        End If
      Case 40  '[↓]:高度を上げる (スピードダウン)
        If flag = False Then
          mz.SwitchToNumber (mz.Extract - downval)
            If mz.Extract < -1 Then
              mz.SwitchToNumber -1
              pitang.SwitchToNumber 5.7
            Else
              ms.SwitchToNumber (ms.Extract + 0.01)
                If KeyData < 0 Then             'ボタンUP時カメラのアングルを変更する
                  pitang.SwitchToNumber 0
                Else
                  pitang.SwitchToNumber 6.2     '目線にカメラのアングルを変更する
                End If
            End If
        End If
      Case 37  '[←]:左へ移動
        If flag = False Then
          If mx.Extract < 0.4 Then mx.SwitchToNumber 0.4
          If KeyData > 0 Then     'コックピットイメージを回転させる
            mx.SwitchToNumber (mx.Extract - 0.01)
              If mx.Extract > 0.4 Then
                pitrot.SwitchToNumber 0.3
              Else
                pitrot.SwitchToNumber 0
              End if
          Else
            pitrot.SwitchToNumber 0
          End If
        End If
      Case 39  '[→]:右へ移動
        If flag = False Then
          If mx.Extract > 1.4 Then mx.SwitchToNumber 1.4
          If KeyData > 0 Then     'コックピットイメージを回転させる
            mx.SwitchToNumber (mx.Extract + 0.01)
              If mx.Extract < 1.4 Then
                pitrot.SwitchToNumber -0.3
              Else
                pitrot.SwitchToNumber 0
              End if
          Else
            pitrot.SwitchToNumber 0
          End If
        End If
    End Select
  End Sub
</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

★ “Not for sale” っていうのは，「貴重すぎて値段がつけられません」というのと「売る価値もないのでタダで差し上げます」との大きく二つ考えられますが，実際はどうなんでしょう。(兵庫県神戸市・**JANDY18歳**) ……**【影：それは，自分の判断でいいでしょう。】**

(超大作プログラム)　リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net/)からダウンロードしてお楽しみください。

# Windows(Visual C++ V6.0)用



内容は本格派のシミュレーション

# 捕獲屋(ほかくや)さん

二宮　裕志（19歳）

## 内容・操作方法

わかりやすく言うと，戦略シミュレーションゲームで，すべてマウス操作です(ESCキーは例外)。

**左クリック**：決定，
**右クリック**：キャンセル，
**ESCキー**：ゲーム強制終了，です。

## ファイル説明

ソースについて.txt：ソースをコンパイルするときの注意
Capture.exe：ゲーム本体
Manual.exe：説明書
Tool.exe：開発ツール
Capture7.lzh：DirectX7.0用のCapture.exe が入った圧縮ファイル
¥Data：ゲームのデータ
¥Grp：グラフィック(BMP)
¥Music：曲(MIDI)
¥Sound：効果音(WAV)
¥Source：ソース

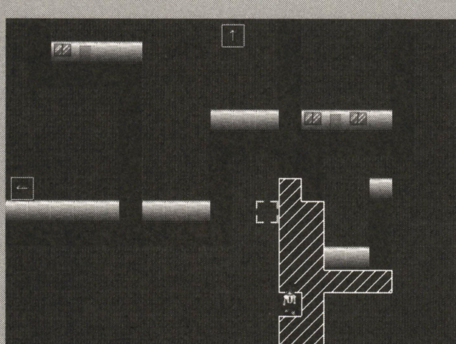
## 注意点

動作環境はMicrosoftWindows95/98DirectX8.0以上です。DirectAudioを使用しているので，DirectX7.0では曲が鳴りません（動作はします)。

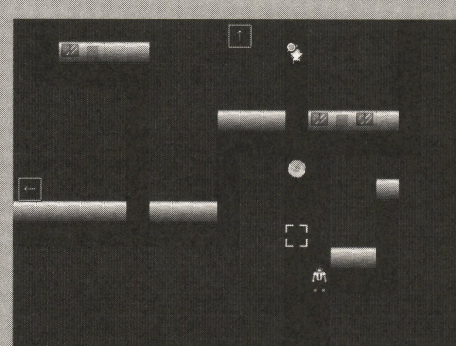
解像度は，ゲーム本体，説明書が640×480 HighColor以上，ツールは解像度1024×768以上です。

DirectX8.0の場合，もしかしたらウィンドウモードにするとサーフェースがめちゃめちゃになるかもしれません。その場合，一度バトルしてからタイトルに戻ると直っているかもしれません。

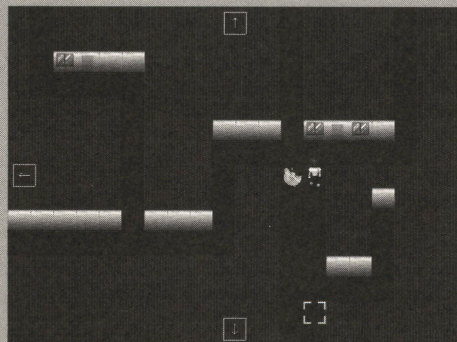
重要なのは，勝手に右クリックが押しっぱなしと認識されてしまう現象です。バトル中に，メニューがクリックできな



《写真1》本格的な作りのシミュレーション



《写真2》内容は餌を配置したあと隠れて



《写真3》見つからないように，正面以外から接近して




《写真4》敵を捕獲すればクリアというゲーム



★大学教授兼プロ棋士の飯田弘之五段に「将棋のプログラム」についてインタビューしてほしい。（千葉県山武郡・野潤上14歳）……
【つぐ美：将棋好きなのね。私もたまにするわよ。影：つぐ美ちゃんを編集者兼女流棋士としてデビューさせる会を結成しました。】

くなる現象が起こるかもしれません(とくに最初よくある)。その場合，一度右クリックすると直るようです。

## 苦労点

new&deleteの嵐，解放失敗かなり苦戦（とくに思考ルーチンあたりに)。

DirectMusic(Audio)を使用すると1トラック目が再生されてないのに気づくまで2か月ぐらいかかりました。なんでそうなるのか今でも謎です。そのせいで曲が適当です（責任転嫁)。

まず，どうしたら「動物捕まえる」ってだけのことを，ゲームっぽくできるか悩みました(それを言ったらなぜにこんなのを題材にしたかも悩む)。

《第1表》ゲームの流れ

起動
↓
ロゴ
↓
タイトル
↓
デモ前半
↓
バトル：ここがゲームの中核
↓
デモ後半
↓
キャンプ
↓
デモ前半へ戻る

※バトルではメニューを使用するオーソドックスな戦略シミュレーションの形態です。全10話。

### CHECKER FLAG

**影**：画面表示などは，かなりの出来映えで文句ナシなんだけど，ゲームとしては,チョット遊びにくい印象だなぁ～。

**編**：文による説明だけでなく，見本となるオートプレイがあると，すぐに内容が把握できると思ったのですが。「百聞は一見にしかず」と言うように，ゲームもプレイを見たほうが理解しやすいです。

**Dr.D**：じゃが，影クンも触れておるように，画面表示や操作面などの完成度は非常に高い作品じゃ。本格的なゲームを作ろうとしている人は，ぜひともリスト面にも注目してほしい。

**つぐ美**：また，ベーマガへの投稿の場合は，できるだけ本文中でゲームの説明を書いてください。細かなツールの使用法などは仕方ないけど，「遊びかたは，ダウンロードファイルに収録されているヘルプやマニュアルを読んで」というのは遠慮してね。ここに掲載している文章だけじゃ，ゲーム内容が読者に全然伝わらないわ。

《第2表》解説すべきシステム

| | |
|---|---|
| **能力値** | |
| ＳＰ | スキルポイント。スキルを使用する際に使用。 |
| 移動力 | 値分だけ移動できる。 |
| 反応 | この距離に進入すると気づかれる。 |
| 視界 | 向いている方向のみ，この距離に侵入すると気づかれる。 |
| 食欲 | エサを食う力。 |
| **バトルシステム** | |
| 捕獲 | 唯一の攻撃らしき方法。 |
| エサ | 敵をおびき寄せるための手段。 |
| 察知 | 敵の反応・視界範囲に侵入すると発生。敵が逃走モードに。 |
| 発見 | 敵の「行動後」に，視界範囲にエサがあると発見する。敵がエサモードに。 |
| 逃走モード | 大体，敵が別のルートで逃げる。 |
| エサモード | 大体，敵がエサに向かってくる。 |
| 勝利 | 敵をすべて捕獲する。 |
| 敗北 | 大体の場合，敵に逃げられると敗北（特定の位置に到達されると)。 |

※大体というのは、一部そうでない場合があるということ。総ターン数はまったくゲームに影響しません。分岐も何もありません。

HPつけて，攻撃なんかして弱ったところ…ってそれじゃポケモン！(とも，考えていた)。視界というものをどう再現するか悩みました。

しかし最大に苦労したのは配置。パズル的…プレイヤー初期位置も重要，敵も，能力値1ずれても，思考ルーチンをちょっと変えても攻略できなくなってしまったり，簡単になりすぎたり。

またマップにしても障害物がひとつ有る無しで，大分変わってしまう。結局は，たいしたものはできなかったのですが(結局おのれの知能の低さの現れか…)。

## 説明書について

もう少し細かい説明はManual.exe(説明書)に入っています。

ちなみにグラフィックは¥Grpからのもの，単体で動いているわけではないので，他のフォルダに移動すると動きません。内容はツールで変更可能です。

## ツールについて

操作方法は，書き始めたら十数ページとかで終わるようなものではないので説明はしません。エラーが発生しても責任は持てません。やり方によっては発生します。

- **キャラツール**：ユニットを作成する(敵味方問わず)。
- **マップツール**：マップ素材を作る(ステージ以外にデモ用のマップなど)。
- **シナリオツール**：ステージを作成する(主に敵のこと)。
- **デモツール**：ステージ前後のデモを作成する（他にロゴ・スタッフロールなど）
- **マニュアルツール**：説明書を作る。
- **ツール補助ツール**：各ツールのレイアウトの調整などをする。
- **試験実行**：ロゴ以外を実行する。
- **終了**：ツールを終了する。

## 攻略法

①敵は決まった行動パターンを持っている。それを知ればどう動けばいいかわかるはず。
②エサの配置場所はよく考えよう。
③最終手段は，無理矢理近づきクイックセーブ&ロードを使用して成功するまで捕獲を繰り返す。
④2面以外は一度も気づかれずにクリアできます。

## その他

一度クリアしたら二度もやる気は起きないかもしれません？またこのゲームは，そもそもゲームとして面白いのか，かなり不安。捕獲した動物をコレクションするとか，罠の作成とか色々付け加えられる要素はあったのですが，1か月あたりで終わらせるはずが，2か月経過してしまったので終わりにしました（気になったら作り直すけど)。シナリオへぼいとか突っ込まないで。それにしても早くメールアドレスがほしいなぁあ…。

**★参考にしたもの**

・電波新聞社　マイコンＢＡＳＩＣマガジン2001年02月号「ＧＥＴ＆ＲＵＮ」
・I/O別冊　DirectX8実践プログラミング(Direct Audioの部分のみ)

## Dr.Dのワンポイント解説

**Dr.D**：画面の演出面が目に付く本作品。その中でも，各場面の切り替えに使われているフィルタ処理のリストに注目しよう。ここでは，実際に描画を行うリストの１部分を掲載してみた。描画方法としては，切り替え時に表示される黒い菱形は，じつはbmpで１コマずつ用意されており，その画像の転送範囲を設定し，blt命令で転送することで，アニメーション形式で画面が切り替わるように見える，という方法じゃな。

**《リスト１》フィルタ処理部分のリストの１部**

```
●Pick Up LIST

BOOL    DrawingFillterClass::Fillter(LPDIRECTDRAWSURFACE &Mat,    //フィルタのサーフェイス
                                     LPDIRECTDRAWSURFACE &dds )    //描画先
{
        if( DFilling == false ) return  FALSE;


        RECT    rcD,    rcS;
        int     tmp,
                Dx,Dy,Dw,Dh,
                Sx,Sy,Sw,Sh;
        DDBLTFX                 DDBfx;
        DDBfx.dwSize = sizeof(DDBfx);
        int flgx = 0 ,flgy = 0;

        //--------------------------------------------------
        //      1.画面にy初期いちから横並べで描画(幅まで)
        //              FirstX,Yのいちから、XSumYSum回数描画
        //--------------------------------------------------
        for( register y = 0 ; y < YSum ; y ++ ){
                for( register x = 0 ; x < XSum ; x ++ ){
                        Sx = rcMat.left;                //もと
                        Sy = rcMat.top;
                        Sw = rcMat.right - rcMat.left;
                        Sh = rcMat.bottom- rcMat.top;

                        Dx = fNowX+x * Sw;      //描画
                        Dy = fNowY+y * Sh;
                        Dw =  Sw;
                        Dh =  Sh;

                        SetRect( &rcD, Dx, Dy, Dx+Dw, Dy+Dh);
                        SetRect( &rcS, Sx, Sy, Sx+Sw, Sy+Sh);
                        InSideRect( ( GrpXRevers == false ? 0 : 1 ), rcFill, &rcS, &rcD);

(途中省略)

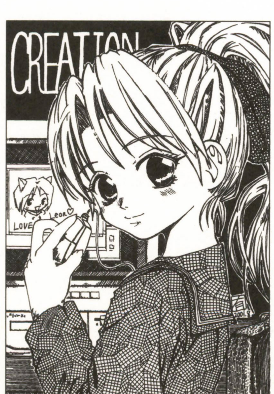
                        //アニメタイル系
                        else{
                            //Backサーフェイスに描画
                            lpDDSBack->Blt( &rcD, Mat,&rcS,DDBLT_KEYSRC | DDBLT_WAIT,0);
                            //なんらかの無駄処理を突っ込んで置かないとうまく動かない ( いまだ謎
)

                            DDSGetPixel( &dds, 0,0);
                        }
```




評…アルバイトで使っている着ぐるみもOK？（影）

by 比呂改め朱目




評…ポニーテールって，たまにするのよ（つぐ美）

by 鈴木有人




評…しっかり自分のホームページの宣伝してるぞ（影）

by ARAI




評…ひそかにリリアのファンだったりする私（影）

by Mink



★いつも思うがどっか似ている「X68K」と「PS2」。（北海道江別市・いまだに68　32歳）……【編：そのネタは，PS2が発売されたときにベーマガでもやりましたね。2000年5月号に「偽X68000」としてPS2をX68000に似せた写真を掲載しています。笑えます。】



投稿作品をiアプリで再現

# iSCROPE

（アイ スクロープ）

仁井内 明（32歳）



## 内容

7年ほど前，X68000用で掲載されたSCROPEのiアプリ対応版。というわけで，iSCROPEです。

ロープと爆弾を使ってブロックを登っていく縦スクロールのアクション・ゲームで，BOYを操りROPEを伸ばしたり，爆弾でブロックを破壊しながら，どこまで上に登れるか挑戦しましょう。

## キャラクター

・**BOY**：主人公です。

・**GIRL**：どうもBOYを避けているような気まぐれな女の子です。でも元気をくれます

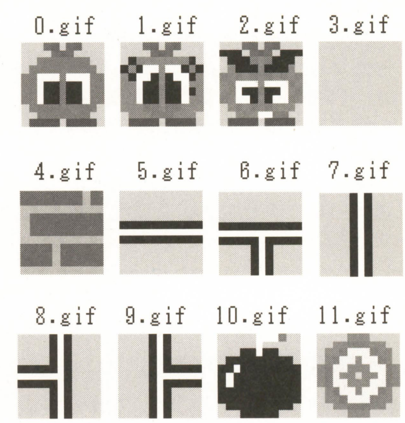
・**WARU**：しつこく追いかけてきて元気を奪います。

・**BLOCK**：爆弾で破壊できます。

・**ROPE**：上と左右に伸ばせます。

・**BOMB**：10カウントで爆発します。

・**FIRE**：ROPEに燃え移ります。




0.9f〜11.9fの画像

## 遊びかた・ルール

方向キーで移動します。ただし，ロープにつかまってないか，下にブロックがなければ落下します。

ソフトキー１＋左，右，上を同時に押すとロープを伸ばせます。ただし，結べない条件がいくつかあります。

ソフトキー１＋下を同時に押すと爆弾を置きます。10カウントで爆発し，上下左右のブロックを破壊できます。炎はロープに燃え移るので気をつけてください。

BOYが画面下まで落下するか，元気(power)が0になるとGAME OVERです。

SCOREは画面を１段スクロールさせると1点プラスされます。それ以外にSCOREは増えません。つまり登った段数がそのままSCOREとなります。

powerに関しては，以下のとおりです。

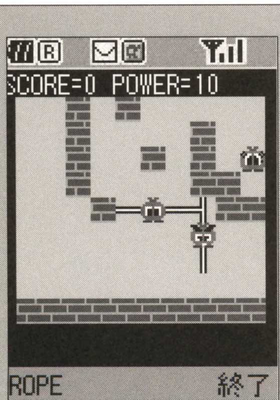
・BOYが炎やWARUに触れると power：－1

・BOYがGIRLに触れるとpower：＋1

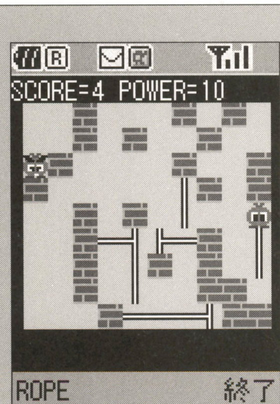
・WARUやGIRLは妖精なので触れると消えてしまいます。

・WARUを炎で消すとpower：＋2

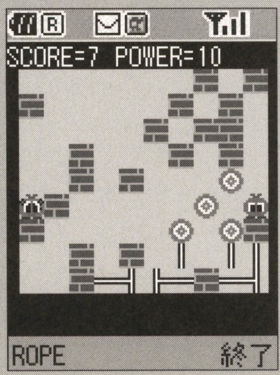
・GIRLを炎で消してしまうと power：－2



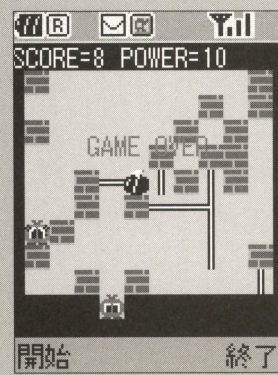

《写真1》昔ながらの読者には懐かしの作品




《写真2》ロープを巧みに使っていく




《写真3》場所によっては爆弾を使用




《写真4》敵に当たったり、下に落下するとダメ

★うちの親曰く「ブロードバンドって何人グループ？」。ブロードバンドを「有名な楽団（バンド）」だと思っている。（新潟県新発田市・canopy19歳）……【影：ちゃんと説明してあげたでしょうね。「幅の広い本皮ベルト」だって。つぐ美：しっかり教えてあげてね。】

## CHECKER FLAG

**編：**懐かしの作品をiアプリへ移植した作品です。世間には数多くのiアプリが出回っている中，こういった過去の移植作品は，ベーマガ投稿iアプリの特徴の一つになるかもしれませんね。
**影：**でも，そのままでの移植でなく，今だからこそ追加できる機能もあったんじゃないかな？移植作品でも，その作者ならではの個性は忘れないでほしいところだ。
**Dr.D：**画面描画やロープなどの判定は，配列に仮想画面を作っておき，キー操作などに対応させてその配列を書き換え，最後にその配列の値に合わせたマップ描画を行うという形じゃな。自キャラや出した直後のロープの動きなどは，drawImageで描画する形じゃ。
**つぐ美：**ウェイト調整機能があるので，エミュレータでも適切な速度で遊べますね。

SCORE100点毎に1ラウンドクリアとなり，難易度が上がります。ブロックの出現する確率が増えるので，なかなか進めなくなります。ボーナスとしてpower：＋5。

## プログラムのしくみ

オブジェクト指向を無視した無限ループ型の流れになっています。LPをタイミングベルトとして各キャラクターのスピードに差をつけています。場面はP変数によってオープニング，ゲーム・オーバー，クリア，ゲーム中の四シーンにわかれていて，switchで振り分けています。

## プログラム入力方法

ベーマガ連載のｉアプリ制作などを見ながらjavaのプログラミング環境を整えてください。さらにGIFデータを作成できるグラフィックソフトを用意してください。これで，二つのソースファイルの入力と12個のGIFデータを作成します。コンパイルすればエミュレータで遊べます。携帯で遊ぶ場合は，リンクを張ったhtmlとjarとjamを自分のHPにアップロードしたあと，携帯へダウンロードして遊びます。

ｉアプリのプログラミング環境はいろいろあるのですが，今回使用したGaDekが簡単無料でおすすめです。http://g-applli.net/に行ってみてください。

## 画像データ

画像データは，12個で11×11のサイズで作ってください。中途半端ですが，左右対称のキャラクターにしたかったのでこうなりました。おかげでブロックのパターンがずれる結果になりました。

## 改造のアドバイス

X68k版では下から水が満ちてきたり，エンディングがあったのですが，今回は省きました。iSCROPEの名前にちなんで，アイスを食べると何か起こるなど，いろいろ発展できると思います。じつは一瞬炎を伝って移動できてしまうバグがあるのですが，忍法だと思ってそのままにしてあります。

ハードの性能によってスピードが違うので，初期設定のpaceで調整してください。1が最速で大きくすると遅くなります。携帯用は1，パソコンはcpuのクロック/50くらいにしてみてください。BOMBのカウンタが0.5秒位で表示されればちょうどいいと思います。

使用携帯機種は方向キーが上下左右に独立のタイプがいいと思います。エミュレーターではP503i,F503iで試しました。

## 最後に

はじめてのｉアプリなので，javaの勉強しながらとりあえず移植から始めました。ゲームを作るのは久しぶりなんですが，最近の言語は論理式が使えなくて，if文を多用する結果になり，ちょっと残念です。ブロックはランダムに出している割にいいバランスで出現するし，燃え移る炎の動きが気に入っているのですが…

### 《リスト１》iscrope.javaのプログラム・リスト

```
// ------『iSCROPE』(1/2) ------
// filename="iscrope.java"
// by Nip (仁井内 明) 2001.06.16～2001.07.07
// H P :http://ww4.tiki.ne.jp/~nip-/
// mail:nip-@mx4.tiki.ne.jp

import com.nttdocomo.ui.*;

public final class iscrope extends IApplication
{
  public void start()
  {
    iscropeCANVAS c=new iscropeCANVAS();
    Display.setCurrent(c);
    c.exe();
  }
}
```

### 《リスト２》iscropeCANVAS.javaのプログラム・リスト

```
// ------『iSCROPE』(2/2) ------
// filename="iscropeCANVAS.java"

import com.nttdocomo.ui.*;
import java.util.*;

//■■■■■■ メインプログラム ■■■■■■
final class iscropeCANVAS extends Canvas {
  public void paint(Graphics g) {}
  void exe() {
    //====== 初期設定 ======
    int pace;             // 全体の実行速度調節(pace):1が最速です。
    int i,j,k,P,LP,bmc;   // 雑用(i,j,k),ﾊﾟﾀｰﾝ(P),ﾀｲﾐﾝｸﾞ(LP),BOMBｶｳﾝﾀ(bmc)
    int myX,myY,bmX,bmY;  // BOYの位置(x,y),BOMBの位置(x,y)
    int giX,giY,waX,waY;  // GIRLの位置(x,y),WARUの位置(x,y)
    int S,pw,Lv,mati;     // SCORE(S),POWER(pw),レベル(Lv),ROPE延縮時ｽﾘｰﾌﾟ(mati)
    P=0    ; S=0;
    pw=0   ; LP=0;
    bmc=0  ; Lv=0;
    myX=0  ; myY=0;
```

★文化祭の華になろうと気合いのこもったゲームを作成。この調子でベーマガにも投稿しようと思います。（滋賀県守山市・じゃーがポテト13歳）……【**つぐ美：**で，その評判はどうでした？華になれました？**編：**ぜひ投稿もお願いしますね。お待ちしています。】

```
bmX=0  ; bmY=0;
giX=0  ; giY=0;
waX=0  ; waY=0;
pace=5 ; mati=0;

// 画面の配列
// image=0:BOY,1:GIRL,2:WARU,3:BACK,4:BLOCK,5:R_LR,6:R_LRD
//       7:R_UD,8:R_UDL.9:R_UDR,10:BOMB,11:FIRE
int[][] MAP={{4,3,3,3,3,3,3,3,3,3,3,4},
             {4,3,3,3,3,3,3,3,3,3,3,4},
             {4,3,3,3,3,3,3,3,3,3,3,4},
             {4,3,3,3,3,3,3,3,3,3,3,4},
             {4,3,3,3,3,3,3,3,3,3,3,4},
             {4,3,3,3,3,3,3,3,3,3,3,4},
             {4,3,3,3,3,3,3,3,3,3,3,4},
             {4,3,3,3,3,3,3,3,3,3,3,4},
             {4,3,3,3,3,3,3,3,3,3,3,4},
             {4,3,3,3,3,3,3,3,3,3,3,4},
             {4,3,3,3,3,3,3,3,3,3,3,4}};

//====== 乱数の定義 ======
Random rand = new Random();

//===== 画像データの読込みクラス ======
Graphics g=getGraphics();
Image[] image = new Image[13];
try {
  MediaImage m;
  for (i=11;i>=0;i--){
    m = MediaManager.getImage("resource:///" + i + ".gif");
    m.use(); image[i] = m.getImage();
  }
} catch (Exception e) {}

// ■■■■■■ 無限ループ ■■■■■■
while(true){
  k=getKeypadState();
  if ((1<<Display.KEY_SOFT2&k)!=0) P=0;  //SOFT2で終了
  LP--;if (LP<=0) LP=105;
  switch(P){
    case 0: // □□□□□□ オープニング □□□□□□
      setSoftLabel(Frame.SOFT_KEY_1,"開始");
      setSoftLabel(Frame.SOFT_KEY_2,"終了");
      g.lock();
      g.setColor(g.getColorOfName(g.BLACK));
      g.fillRect(0,0,getWidth(),getHeight());
      g.setColor(g.getColorOfName(g.WHITE));
      g.drawString("iSCROPE",39,60);
      g.drawString("speed="+pace,39,80);
      g.unlock(true);
      if (LP%5==0)
      if ((1<<Display.KEY_SOFT1&k)!=0){
        P=3   ; Lv=2;
        S=0   ; pw=10;
        myX=6 ; myY=8;
        waX=0 ; waY=10;
        giX=0 ; giY=10;
        bmX=0 ; bmY=10;
        mati=3000*pace;
        for(j=8;j>=0;j--){
          for(i=10;i>=1;i--) MAP[j][i] = 3;
          if (j<6) for(i=Lv;i>=0;i--) MAP[j][(rand.nextInt()>>>1)%10+1]=4;
        }
        for(i=10;i>=1;i--) MAP[9][i]=4;
        setSoftLabel(Frame.SOFT_KEY_1,"ROPE");
        }
        else if((1<<Display.KEY_LEFT &k)!=0 && pace>1)    pace--;
        else if((1<<Display.KEY_RIGHT&k)!=0 && pace<1000) pace++;
        else if((1<<Display.KEY_SOFT2&k)!=0)
          IApplication.getCurrentApp().terminate();
      break;

    case 1: // □□□□□□ ゲームオーバー □□□□□□
      setSoftLabel(Frame.SOFT_KEY_1,"開始");
      g.lock();
      g.setColor(g.getColorOfName(g.RED));
      g.drawString("GAME OVER",33,50);
      g.unlock(true);
      if ((1<<Display.KEY_SOFT1&k)!=0) P=0;
      break;

    case 2: // □□□□□□ クリア □□□□□□
      setSoftLabel(Frame.SOFT_KEY_1,"次へ");
      g.lock();
      g.setColor(g.getColorOfName(g.BLUE));
      g.drawString("===CLEAR!!===",21,48);
      g.unlock(true);
      if ((1<<Display.KEY_SOFT1&k)!=0){
        P=3; pw=pw+5; Lv++;
        setSoftLabel(Frame.SOFT_KEY_1,"ROPE");
      }
      break;

    case 3: // □□□□□□ ゲーム □□□□□□
      // ====== BOY ======
      if(LP%pace == 0){
        if ((1<<Display.KEY_SOFT1&k)!=0){                //"①"を押しながら
          if ((1<<Display.KEY_DOWN&k)!=0 && bmY>9){    //"↓"BOMBを置く
            bmX=myX; bmY=myY; bmc = 10;
            MAP[bmY][bmX]=10;
          }
          else if ((1<<Display.KEY_UP&k)!=0){          //"↑"ROPE上
            for(i=myY;(MAP[i-1][myX]==3)&&(i>1);i--){
              MAP[i-1][myX] = 7;
              g.lock();
              g.drawImage(image[7],myX*11-6,i*11-10);
              g.unlock(true);
              for(j=mati;j>=0;j--);
            }
            for(j=mati;j>=0;j--);
            if(i==1){
              for(i=i;i<myY;i++){
                g.lock();
                g.drawImage(image[3],myX*11-6,i*11+1);
                g.unlock(true);
                MAP[i][myX] = 3;
                for(j=mati;j>=0;j--);
              }
            }
            else if(MAP[i-1][myX]==5) MAP[i-1][myX] = 6;
          }                                               //"→"ROPE左
          else if ((1<<Display.KEY_LEFT&k) != 0 && MAP[myY][myX]!=9){
            for(i=myX;(MAP[myY][i-1]==3)&&(i>1);i--){
              MAP[myY][i-1] = 5;
              g.lock();
              g.drawImage(image[5],i*11-17,myY*11+1);
              g.unlock(true);
              for(j=mati;j>=0;j--);
            }
            for(j=mati;j>=0;j--);
            if(MAP[myY][i-1]==8 || i==1){
              for(i=i;i<myX;i++){
                g.lock();
                g.drawImage(image[3],i*11-6,myY*11+1);
```

```
          g.unlock(true);
          MAP[myY][i] = 3;
          for(j=mati;j>=0;j--);
        }
      }
      else if(i!=myX){
        if(MAP[myY][i-1]==7) MAP[myY][i-1] = 9;
        if(MAP[myY][myX]==7 || MAP[myY][myX]==3) MAP[myY][myX] = 8;
      }
    }                                              //"←"ROPE左
    else if ((1<<Display.KEY_RIGHT&k)!=0 && MAP[myY][myX]!=8){
      for(i=myX;(MAP[myY][i+1]==3)&&(i<10);i++){
        MAP[myY][i+1] = 5;
        g.lock();
        g.drawImage(image[5],i*11+5,myY*11+1);
        g.unlock(true);
        for(j=mati;j>=0;j--);
      }
      for(j=mati;j>=0;j--);
      if(MAP[myY][i+1]==9 || i==10){
        for(i=i;i>myX;i--){
          g.lock();
          g.drawImage(image[3],i*11-6,myY*11+1);
          g.unlock(true);
          MAP[myY][i] = 3;
          for(j=mati;j>=0;j--);
        }
      }
      else if(i!=myX){
        if(MAP[myY][i+1]==7) MAP[myY][i+1] = 8;
        if(MAP[myY][myX]==7 || MAP[myY][myX]==3) MAP[myY][myX] = 9;
      }
    }
  }//"①"を押さずに
  else if((1<<Display.KEY_UP & k)!=0 && MAP[myY-1][myX]>5 && myY>1){
    if(myY>5) myY--;
    else {
      S++; waY++; giY++; bmY++;
      for(j=9;j>=1;j--)
        for(i=10;i>=1;i--)
          MAP[j][i] = MAP[j-1][i];
      for(i=10;i>=1;i--) MAP[0][i] = 3;
      for(i=Lv;i>=0;i--) MAP[0][(rand.nextInt()>>>1)%10+1] = 4;
    }
  }
  else if((1<<Display.KEY_DOWN &k)!=0&& MAP[myY+1][myX]!=4 && myY<9)
    myY++;
  else if((1<<Display.KEY_LEFT &k)!=0&& MAP[myY][myX-1]!=4 && myX>1)
    myX--;
  else if((1<<Display.KEY_RIGHT&k)!=0&& MAP[myY][myX+1]!=4 && myX<10)
    myX++;
  if(myY==giY && myX==giX) {pw++ ; giY=10;}
  if(myY==waY && myX==waX) {pw-- ; waY=10;}
  if(MAP[myY+1][myX]!=4 && MAP[myY][myX]==3 && myY<10) myY++;
}

if(LP%(pace*7) == 0){
  // ====== WARU ======
  if (waY<10) {
    if(MAP[waY-1][waX]>5) waY--;
    else if(MAP[waY][waX]==3 && MAP[waY+1][waX]!=4) waY++;
    else if(waX<myX && MAP[waY][waX+1]!=4) waX++;
    else if(waX>myX && MAP[waY][waX-1]!=4) waX--;
  }
  else if(waY>9 && (rand.nextInt()>>>1)%15==0) {
    waX = (rand.nextInt()>>>1)%9+1;
    if (MAP[1][waX]==3 ) waY=1;;
  }
```

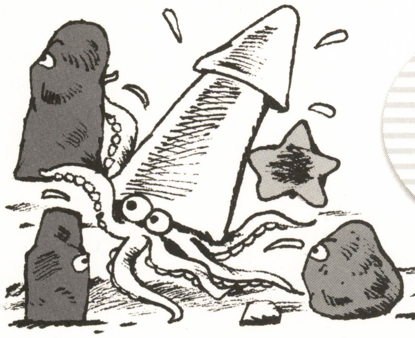
```
        // ====== GIRL ======
        if (giY<10) {
          if(MAP[giY-1][giX]>5) giY--;
          else if(MAP[giY][giX]==3 && MAP[giY+1][giX]!=4) giY++;
          else if(giX<myX && MAP[giY][giX-1]!=4 && giX>1) giX--;
          else if(giX>myX && MAP[giY][giX+1]!=4 && giX<10) giX++;
        }
        else if (giY>9 && (rand.nextInt()>>>1)%25==0) {
          giX = (rand.nextInt()>>>1)%9+1;
          if(MAP[1][giX]==3) giY=1;
        }
      }

      // ====== FIRE ======
      if(giY<10) if(MAP[giY][giX]==11) {pw=pw-2;giY=10;}
      if(waY<10) if(MAP[waY][waX]==11) {pw=pw+2;waY=10;}
      if(LP%(pace*3) == 0 ){
        if(MAP[myY][myX]==11) pw--;
        for (j=9;j>=1;j--)
          for (i=10;i>=1;i--)
            if(MAP[j][i]==11){
              MAP[j][i]=3;
              if (MAP[j-1][i]>6) MAP[j-1][i] = 0;
              if (MAP[j+1][i]>6) MAP[j+1][i] = 0;
              if (MAP[j][i-1]== 5 || MAP[j][i-1]== 6 || MAP[j][i-1]==9)
                MAP[j][i-1] = 0;
              if (MAP[j][i+1]== 5 || MAP[j][i+1]== 6 || MAP[j][i+1]==8)
                MAP[j][i+1] = 0;
            }
      }

      // ====== BOMB ======
      if(LP%(pace*5) == 0 && bmY<10){
        bmc--;
        if (bmc == 0) {
          MAP[bmY-1][bmX] = 11;
          MAP[bmY+1][bmX] = 11;
          MAP[bmY][bmX-1] = 11;
          MAP[bmY][bmX+1] = 11;
          bmY=10;
        }
      }

      // ====== 画面表示 ======
      g.lock();
      g.setColor(g.getColorOfName(g.BLACK));  // 背景色の指定
      g.fillRect(0,0,getWidth(),getHeight()); // 画面全体の塗りつぶし
      for (j=9;j>=1;j--)
        for (i=10;i>=1;i--) {
          if(MAP[j][i]==0) MAP[j][i] = 11;
          g.drawImage(image[MAP[j][i]],i*11-6,j*11+1);
        }
      if(waY<10) g.drawImage(image[2],waX*11-6,waY*11+1);      //WARUの表示
      if(giY<10) g.drawImage(image[1],giX*11-6,giY*11+1);      //GIRLの表示
      g.drawImage(image[0],myX*11-6,myY*11+1);                 //BOYの表示
      g.setColor(g.getColorOfName(g.WHITE));
      g.drawString("SCORE=" + S + " POWER="+pw ,0,11);         //SCOREの表示
      if(bmY<10) g.drawString(" "+(bmc-1),bmX*11-9,bmY*11+11);//BOMBの数字
      if (myY>9 || pw<1 ) P=1;                                 //GAME OVER
      else if (S>=(Lv-1)*100 ) P=2;                            //クリア
      g.unlock(true);
      break;
    } // switch
  } // while
 } // exe
} // Canvas
```

# iアプリ用



iアプリでポケコンの画面？

# イカクン ガンバレ！ for i-mode

りっちもんど（28歳）

## はじめに

このゲームは十数年前，私がポケコンで作ったものを移植したものです。完全移植を目指してみました（笑）。

## 内容

ワンキー横スクロール・ゲームです。とにかく前へ進んでいってください。画面を横向きに使用するので，携帯を横にして持つ必要があります。

## 操作方法

使用するのは，方向，決定，数字ボタンのうち，お好きなもの一つです。

プログラムを開始したあと，ボタンを押すとゲーム開始です。イカクンはボタンを押すと上に，放すと下に移動します。

終了するのはソフト１キーです。

## ルール

障害物が画面の右側からやってきます。障害物に押されて，イカクンが画面から押し出されるとゲーム・オーバーです。途中，ハートマークが出てきますが，これを取るとPOWERが一つ増えます。これ一つで，障害物を一つ壊せます。障害物の壊しかたは，ただ単にぶつかるだけです。POWERは最大で五つまでストックできます。

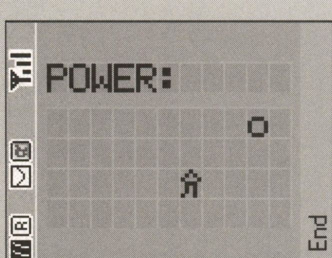
イカクンの当たり判定は，前方３マスにしかありません(**第１図参照**)。真上，真下は障害物があっても移動できます。ただし，ハートマークも取れません。

## 得点について

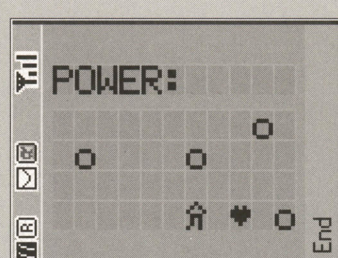
一つ前へ進めば→＋１点。

ボタンを押していたのを離した，または離していたのを押した→＋２点。

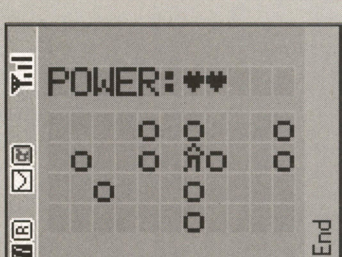
ハートマークを取った→＋10点。




《写真１》ポケコンの画面を再現？




《写真２》ルールは単純。敵を避けるだけ




《写真３》後半は，かなりキビシイ状況も




《写真４》ハートをうまくとることが重要

## CHECKER FLAG

**影**：画面を横に使うというアイデアがいいなぁ～。ワンダースワンあたりからヒントを得たのかな？そういえば，ドリキャスのVMミニゲームにも，画面を横に使うゲームがあったなぁ。

**つぐ美**：ポケコンの雰囲気を再現する，というのがベーマガ読者の発想らしくていいですね。ただ，gifファイルで，バックの色を作成しないと，このポケコンらしさを再現できないのが面倒ですが。

**Dr.D**：ゲーム内容が単純なため，リストも全体的にコンパクトにまとまっていて，見やすい形式じゃ。ただ，当たり判定などは，人によってはいい加減な設定と感じるかもしれんが？iアプリでウェイト処理を入れているのには驚きじゃね。

★①名古屋にもパ・リーグを！「名鉄チキンランナーズ」で，どう？②タレをかけ，ウナギを食べた，つもりだけ。いつかは食べたい，ひつまぶし夢。（愛知県名古屋市・**最馬場可？歳**）……【**影：彼はいつもハガキにびっちり書いてきてくれますが，掲載は1本だけね。**】

## プログラムについて

ポケコンの処理スピードの遅さを再現するべく，Thread.sleepを使っていますが，ただ単に sleep(400) としただけでは，操作性がかなり悪かったので，sleep(50)を複数回使うことにしました。ポケコンでの先行入力ぎみのキー入力が，意外とうまく再現できていると自己満足してます。

## 感想など

i-JADEではちゃんと当たり判定ができるのに，Dojaでは変になります。何が悪いのかさっぱりわかりません。よって，Dojaは使用不可です（笑）。

実機が手元にないので，確認できてないのも気になる点です。しかも，携帯を横向きにして持って遊ぶ前提なのに，エミュレーターは縦にしか表示できないのも少し困りました。

しばらく，ベーマガから離れていたのですが，最近，また買いだしました。投稿するのは初めてです。でも，近くの本屋には売ってないです。こんなにマイナーな本だったかなぁ（失礼）？

パソコンにBASICが付いてこなくなった頃から，自分でプログラムを組むことがなくなっていたけど，最近，再認識。どんなにショボくても，自分で組んだプログラムがちゃんと動いてくれるのが，パソコンを使っていて一番楽しいね。

| | |
|---|---|
| ○● | ※“分”は自キャラを示す |
| 分● | ●は判定あり。 |
| ○● | ○は判定なし。 |

**《第１図》当たり判定図**



chr5.gif～chr6.gifの画像



chr17.gifの画像



chr0.gif～chr4.gif,chr7.gif～chr16.gifの画像

**《リスト１》Ikakun.javaのプログラム・リスト**

```
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import java.util.*;

public class Ikakun extends IApplication{
  public void start(){
    PhoneSystem.setAttribute(PhoneSystem.DEV_BACKLIGHT,PhoneSystem.ATTR_BACKLIGH
T_ON);
    main m = new main();
    Thread r = new Thread(m);
    r.start();
    Display.setCurrent(m);
  }
}

class main extends Canvas implements Runnable{
  Random rnd = new Random();   // 乱数用
  Image chr[] = new Image[18];   // 各種イメージ
  int m[][] = new int[11][5];   // 仮想画面
  int x, y;   // 自機の座標
  int offsetx, offsety;   // 画面表示補正値
  int key = 0;   // キー入力値
  int oldkey;   // 前回のキー入力値
  int score;   // スコア
  int level;   // 障害物の数の調整
  int heart;   // パワーの数
  int chrmode;   // ゲームの進行状態（兼・自機表示の切替）
  final int gamespeed = 7;   // ゲームスピード

    ///// コンストラクタ

  public main(){
    offsetx = (getWidth()-82)/2;
    if (offsetx<0) offsetx = 0;
    offsety = (getHeight()-120)/2;
    if (offsety<0) offsety = 0;
    getdata();
    myinit(0);
    setSoftLabel(SOFT_KEY_1," End");
  }

    ///// 作画処理

  public void paint(Graphics g){
    g.lock();
    setBackground(g.getColorOfRGB(159,192,44));
    g.clearRect(0,0,getWidth(),getHeight());
    for (int ix=0; ix<=10; ix++){
      for (int iy=0; iy<=3; iy++) g.drawImage(chr[m[ix][iy]],(3-
iy)*15+offsetx,ix*11+offsety);
      if ((ix==10)||(ix<=4)) g.drawImage(chr[m[10-ix][4]],67+offsetx,(10-
ix)*11+offsety);
    }
    if (chrmode>0) g.drawImage(chr[chrmode],(3-y)*15+offsetx,x*11+offsety);
    else g.drawImage(chr[17],15+offsetx,33+offsety);
    g.unlock(true);
  };

    ///// メインループ

  public void run(){
    for (;;){
      repaint();
      scroll();
      if (chrmode==0) matiloop(); else mymove();
```

```
    }
  }

    ///// スクロール処理

  public void scroll(){
    int rd = 0;   // 乱数
    int chk = 0;   //障害物の重複数のカウンタ
    for (int iy=0; iy<=3; iy++){
      for (int ix=1; ix<=10; ix++) m[ix-1][iy] = m[ix][iy];
      rd = Math.abs(rnd.nextInt()%1000);
      if ((rd<level)&&(chk<3)){
        m[10][iy] = 1;
        chk++;
      } else m[10][iy]=0;
      if (rd<5) m[10][iy]=2;
    }
  }
```



★はじめてお便りを出します。この本を友人から借りて読んでます。プログラムのことがたくさん載っていて，とても気に入りました。P.S. 文化祭で僕の作ったゲームが好評でした。（岐阜県多治見市・物理部部長14歳）……【つぐ美：これからは毎月買ってね。】

```
///// 自機の移動と当たり判定

public void mymove(){
  int yy = 0;   // 移動方向
  boolean flag = true;   // 入力受け付けフラグ
  for (int i=0; i<=gamespeed; i++){
    if (flag){
      getkey();
      if (key!=0) yy= -1; else yy = 1;
      if (oldkey!=key){
        flag = false;
        score += 2;
      }
    }
    wait(50);
  }
  oldkey = key;
  y = y+yy;
  if (y==-1) y=0;
  if (y==4) y=3;
  switch (m[x][y]){
    case 1:
      if (heart==0){
        x--;
        if (x==-1){
          gameover();
        }
      } else {
        m[x][y] = 0;
        setheart(false);
      }
      break;
    case 2:
      score+=10;
      m[x][y] = 0;
      if (heart<5) setheart(true);
      break;
  }
  score++;
  level++;
}

///// パワーの増減

public void setheart(boolean flag){
  if (flag){
    heart++;
    m[5+heart][4] = 2;
  } else {
    m[5+heart][4] = 0;
    heart--;
  }
}
///// ウェイト処理

public void wait(int w){
  try {Thread.sleep(w);} catch (InterruptedException e){}
}

///// ゲームオーバー時のキー入力待ち

public void matiloop(){
  for (int i=0; i<=gamespeed; i++){
    getkey();
    if (key!=0){
      myinit(1);
      break;
    }
    wait(50);
  }
}

///// 各変数の初期設定

public void myinit(int flag){
  if (flag<=1){
    for (int iy=0; iy<=4; iy++){
      for (int ix=0; ix<=10; ix++) m[ix][iy] = 0;
    }
  }
  level = 100;
  chrmode = 0;
  m[0][4] = 6;
  if (flag==1){
    chrmode = 3;
    score = 0;
    m[0][4] = 5;
    x = 6;
    y = 0;
    oldkey = 0;
    heart = 0;
  }
}

///// ゲームオーバーの処理

public void gameover(){
  int tmp = 1;   // スコア表示に使用
  x = 0;
  for (int i=1; i<=9; i++){
    chrmode = i%2*3+1;
    repaint();
    wait(300);
  }
  for (int i=6; i<=10; i++) m[i][4] = 0;
  myinit(2);
  String st = Integer.toString(score);
  for (int i=2; i<=st.length(); i++) tmp = tmp*10;
  for (int i=1; i<=st.length(); i++){
    m[6+i][4] = (score/tmp)+7;
    score -= (score/tmp)*tmp;
    tmp = tmp/10;
  }
}

///// キー入力

public void getkey(){
  key = getKeypadState();
  if ((key & (1<<Display.KEY_SOFT1))!=0)
IApplication.getCurrentApp().terminate();
}

///// イメージ読み込み

public void getdata(){
  for (int i=0;i<=17;i++){
    MediaImage c =
MediaManager.getImage("resource:///chr"+Integer.toString(i)+".gif");
    try { c.use(); } catch(ConnectionException e){}
    chr[i] = c.getImage();
  }
}
}
```



★僕の家では，X68Kがいまだに現役です。プログラム送ってもいいですか？（秋田県横手市・**タッカー25歳**）……【**影**：おや，今月どこかでX68Kネタあったなぁ。集中するときは集中するもんだ。**編集長**：影君，仕事も集中して頼むぞ。この校正ミスはなんだ！】

# PC-9801シリーズ互換機(N-88日本語BASIC(86)DOS-BASIC)用



アイス(?)みたいなミサイルで対戦

# メロンヴァー・スイカヴァー

怪盗プー（18歳）

## ゲーム内容

お腹減った星人とハングリー星人の対戦型シューティング・ゲームです。

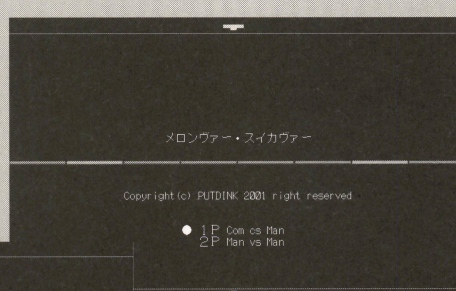
## 遊びかた

相手の陣地に，メロンヴァーやスイカヴァーを押し込むゲームです。

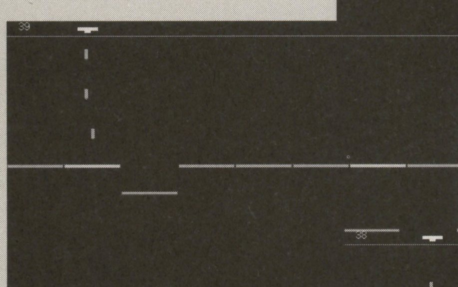
ミサイルが当たるとヴァーが動きます。

ミサイルが連続で当たった場合，移動する距離は累積されます。自分の陣地に向かってくるヴァーにミサイルを当てると，その場で止まります。

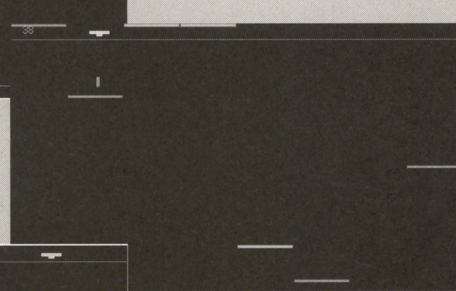
メロンヴァーは移動距離と得点がスイカヴァーの２倍です。



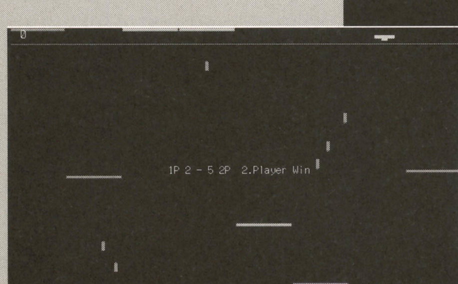

《写真１》対戦型のシューティング



《写真２》真ん中のヴァーを撃ち合って



《写真３》相手の陣地に入れていけばOK




《写真４》ヴァーの色による得点が勝敗の鍵

## 操作方法

**1P／2P選択**：8，2キーで移動，スペース・キーで選択。

**１Ｐの操作**：A，Dキーで移動，スペース・キーでミサイル発射。

**２Ｐの操作**：4，6キーで移動，エンター・キーでミサイル発射。

上が1P，下が2Pです。対COM戦は，下がプレイヤーです。

## 苦労したところ

初めてCOMの制作をしたので，COM制作には苦労しました。ベーマガの様々なプログラミング講座で言われているような「人間だったらどうするか」というのが身にしみました。おかげで，負ける

**CHECKER FLAG**

**つぐ美**：久しぶりのN-88BASIC作品です。投稿の受付けを締め切ったわけではないので，まだまだ投稿募集中ですよ～。PC-98ユーザーさん立ち上がって～。

**Dr.D**：改めて思ったが，こういったN-88BASICなどは，VBなどと違って一から自分で制作していくため，画面構成なども，自然と個性を持った形になるのぉ。VBを使用しているという読者なども，言語自体は参考にならんかもしれんが，このN-88BASICならではの個性は，ゲーム制作などにおいて参考になるはずじゃ。

**編**：ヴァーの色は１色による塗りつぶしではないため，立体感を持った形になっているのが，チョットしたこだわりと言えます。

★OFでは本名を結構ばらされるけど，はじめから本名で書いておけば大丈夫。（福岡市・**前田宣弘27歳**）……【**影**：そういえば過去に，投稿プログラムがペンネームで掲載された後，大学の推薦のために，本人であるという証明をしてあげたことがあったなあ。】

ことはそうそうないにしても，気の抜くことができないCOMができました。

## 改造のアドバイス

メロンヴァー・スイカヴァーの他に，ガリガ○くんなんかをつけるといいかもしれません。ゲーム速度は，行番号1020のSTPで決まっています。小さくするほど速くなります。

## PUTDINKのお話

PUTDINKは山形に本社を持つグループで，会長，副会長，社長，怪盗プー，その他のメンバーで構成されています。

たまたま帰郷した折にN-88BASICで作ろうということになりましたが，ふだんはVB中心でやってます。HPアドレスはこちら。
http://members.tripod.co.jp/putdink/

**《第1表》変数表**

| | |
|---|---|
| BAR(m,n) | ヴァーの画像 |
| TANK(m,n) | 戦車の画像 |
| SHELL(m,n) | ミサイルの画像 |
| PN(m) | 獲得ポイント |
| BT(m) | ヴァーのタイプ(スイカとメロン) |
| BY(m) | ヴァーのY座標 |
| BS(m) | ヴァーの移動方向 |
| BM(m) | ヴァーのミサイル累積 |
| X(m) | 戦車のX座標 |
| SH1(m) | ミサイルの発射判定 |
| SH2(m,n) | ミサイルの位置 |
| SW(m) | ミサイル発射間隔 |
| TIM | プレイ時間 |
| TX | COMの移動先X座標 |
| TYPE | COMの行動タイプ |
| MST | COMのミサイル発射間隔 |
| TTIM | COMの停止時間カウント |

**『メロンヴァー・スイカヴァー』のプログラム・リスト**

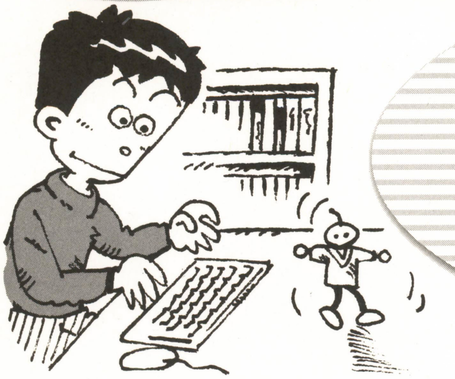
```
1000 '-------------------------【初期設定】-------------------------------
1010 SCREEN 3,0,0,1:CONSOLE 0,25,0,1:WIDTH 80,25:COLOR 7,0,,,2:CLS 3
1020 RANDOMIZE VAL(RIGHT$(TIME$,2)):STP=1500
1030 DIM BAR(50,1),TANK(100,1),SHELL(15,1),PN(2)
1040 DIM BC(7),BT(7),BY(7),BS(7),BM(7),X(1),SH1(10),SH2(10,2),SW(1)
1050 LINE(1, 1)-(79, 5),10,B    :LINE(2, 2)-(78, 4),2,BF
1060 LINE(1,11)-(79,15),13,B    :LINE(2,12)-(78,14),5,BF
1070 LINE(1, 100)-(29,105),1,BF:LINE(11,105)-(19,108),1,BF
1080 LINE(41,100)-(49,104),2,BF:LINE(31,104)-(59,108),2,BF
1090 CIRCLE(5 ,215),2,8,,,,F    :LINE(3 ,203)-(7 ,215),15,BF
1100 CIRCLE(15,206),2,8,,,,F    :LINE(13,206)-(17,217),15,BF
1110 GET(1,  1)-(79,  5),BAR(0,0)   :GET(1,  11)-(79, 15),BAR(0,1)
1120 GET(1,101)-(30,108),TANK(0,0)  :GET(31,101)-(60,108),TANK(0,1)
1130 GET(3,203)-(7 ,217),SHELL(0,0):GET(13,203)-(17,217),SHELL(0,1)
1140 CLS 3:GOSUB *GAME.SET:GOSUB *GAME.START
1150 '-------------------------【入力判定】-------------------------------
1160 *MAIN
1170    FOR I=0 TO 10:IP(I)=255-INP(&HE0+I):NEXT:IP(9)=-IP(9)*(PM=2)
1180    X0=-(PM=2)*((IP(2)=2)-(IP(2)=16)):X1=(IP(0)=16)-(IP(0)=64)
1190    ON PM GOSUB *AI.1
1200      IF X0◇0 THEN P1=0:P2=X0:GOSUB *TANK.MOVE
1210      IF X1◇0 THEN P1=1:P2=X1:GOSUB *TANK.MOVE
1220      IF IP(9)=64  AND SW(0)=0 THEN P1=0:GOSUB *SHELL
1230      IF IP(1)=128 AND SW(1)=0 THEN P1=1:GOSUB *SHELL
1240    FOR A=0 TO 1:FOR B=1 TO 5:ON SH1(A*5+B) GOSUB *SHELL.MOVE:NEXT B,A
1250    FOR A=0 TO 7:ON ABS(BS(A)) GOSUB *BAR.MOVE:NEXT
1260    FOR A=0 TO 1:SW(A)=SW(A)+(SW(A)>0):NEXT A
1270    ON PM GOSUB *AI.2
1280    GOSUB *WAIT.:GOSUB *TIME.COUNT:IF TIM=0 THEN GOTO *GAME.END
1290 GOTO *MAIN
1300 '-------------------------【各種機能】-------------------------------
1310 *GAME.SET
1320    CLS 3:X(0)=300:X(1)=300:TIM=60:TM$=TIME$:LOCATE 1,0:PRINT TIM
1330    RN=INT(RND*2):ROL=(RN=>1)-(RN<1):TX=300+ROL*200:TYPE=1:MST=15
1340    FOR A=0 TO 7
1350      BY(A)=200:BT(A)=(RND*2) MOD 2:PUT(A*80,200),BAR(0,BT(A))
1360    NEXT A
1370    FOR I=0 TO 1
1380      PUT(300,I*370+10),TANK(0,I):LINE(0,I*355+22)-(640,I*355+22),15
1390    NEXT I
1400  RETURN
1410 *TANK.MOVE
1420    IF X(P1)+P2*3<0 OR X(P1)+P2*3>609 THEN RETURN
1430    LINE(X(P1),10+P1*370)-(X(P1)+30,20+P1*370),0,BF
1440    X(P1)=X(P1)+P2*3:PUT(X(P1),10+P1*370),TANK(0,P1),PSET
1450  RETURN
1460 *WAIT.
1470    FOR A=0 TO STP:GOSUB *KEY.SKIP:NEXT:RETURN
1480 *KEY.SKIP
1490    IF INKEY$="" THEN RETURN ELSE RETURN
1500 *TIME.COUNT
1510    IF TM$◇TIME$ THEN TM$=TIME$:TIM=TIM-1:LOCATE 1,0:PRINT TIM
1520 RETURN
1530 *GAME.START
```

```
1540    CLS:PM=1:FOR I=1 TO 4:READ A,B,M$:LOCATE A,B:PRINT M$:NEXT:GOTO 1570
1550    I=0:K$=INKEY$:IF K$="" THEN 1550 ELSE X=(K$="8")-(K$="2")
1560      IF K$=" " THEN GOTO 1590
1570      LOCATE 30,17+PM:PRINT"  ":PM=PM-X*(PM+X=1 OR PM+X=2)
1580      LOCATE 30,17+PM:PRINT"●":GOTO 1550
1590    WHILE I<6
1600        WHILE M$=TIME$:WEND:LOCATE 38,11:PRINT 5-I:M$=TIME$:I=I+1
1610    WEND:CLS
1620 RETURN
1630 *GAME.END
1640    COLOR 2:LOCATE 28,12:PRINT"1P"PN(2)"-"PN(0)"2P";
1650    IF PN(2)>PN(0) THEN M$="1.Player Win" ELSE M$="2.Player Win"
1660    IF PN(2)=PN(0) THEN M$="Draw"
1670    PRINT "  "+M$:COLOR 7:END
1680 *AI.1
1690    MT=MT+1-20*(TIM<30):X0=ROL:IF TYPE=2 THEN TTIM=TTIM+1
1700    IF MT>MST AND BY(-INT(-((X(0)+10)/80))-1)◇0 THEN IP(9)=64:MT=0
1710 RETURN
1720 *AI.2
1730    I=0:II=0:NX=-INT(-(X(0)+10)/80)-1
1740    WHILE I<7 AND II=0
1750      IF NX+I <7 THEN IF 0<BY(NX+I) AND BY(NX+I)<80 THEN II=I :I=8
1760      IF NX-I=>0 THEN IF 0<BY(NX-I) AND BY(NX-I)<80 THEN II=-I:I=8
1770    I=I+1:WEND:I=0
1780    IF II◇0 THEN 1790 ELSE 1800
1790      TX=(NX+II)*80+30:TYPE=1:ROL=SGN(II):MST=5:TTIM=0:GOTO 1900
1800    IF TYPE=2 THEN 1810 ELSE 1880
1810      IF TTIM>25 THEN 1820 ELSE 1900
1820        FOR I=0 TO 7:BB=BB-(BY(I)◇0):NEXT I
1830        IF BB=0 THEN TYPE=2:ROL=0:TTIM=0:GOTO 1900
1840        WHILE LX=0
1850          RN=INT(RND*8):LX=-(BY(RN)◇0)*(RN*80-(RN=0)*30)
1860        WEND:LX=LX+30
1870        ROL=(TX>LX)-(TX<LX):TX=LX:TYPE=1-(ROL=0):TTIM=0:MST=15:GOTO 1900
1880    IF TX-2<X(0) AND X(0)<TX+2 THEN 1890 ELSE 1900
1890      TYPE=2:ROL=0:TTIM=0:MST=10:GOTO 1900
1900 LX=0:BB=0:I=0:II=0:RETURN
1910 *SHELL
1920    I=0:II=0
1930    WHILE I<5:I=I+1:II=-I*(SH1(P1*5+I)=0):I=I-(SH1(P1*5+I)=0)*6:WEND
1940    IF II=0 THEN RETURN ELSE BX=-INT(-(X(P1)+2)/80)-1
1950    IF P1=0 AND BY(BX)>10  AND BY(BX)<41 THEN 1970
1960    IF P1=1 AND BY(BX)>355               THEN 1970 ELSE 1990
1970      FG=-(BS(BX)=0 OR BS(BX)=(P1=1)-(P1=0)):BC(BX)=FG*(BC(BX)+1)
1980      BM(BX)=FG*BM(BX):BS(BX)=FG*((P1=1)-(P1=0)):SW(P1)=5:RETURN
1990    SH2(P1*5+II,0)=X(P1)+10:SH2(P1*5+II,1)=25+335*P1:SH2(P1*5+II,2)=0
2000    SH1(P1*5+II)=1:SW(P1)=5
2010 RETURN
2020 *SHELL.MOVE
2030    AN=((A=1)-(A=0)):XX=SH2(A*5+B,0):YY=SH2(A*5+B,1):SP=5*AN
2040    LINE(XX,YY)-(XX+4,YY+15),0,BF
2050    IF YY+SP=<20 OR YY+SP-(A=0)*15>375 THEN SH1(A*5+B)=0:RETURN
2060    YY=YY+SP:SH2(A*5+B,1)=YY:PUT(XX,YY),SHELL(0,A),PSET
2070    BX=-INT(-(XX+2)/80)-1:BB=BY(BX):FG=-(BS(BX)=0 OR BS(BX)=AN)
2080    IF YY-15*(A=0)>BB-2 AND YY-15*(A=0)<BB+7 THEN 2090 ELSE 2110
2090      BC(BX)=FG*(BC(BX)+1):BM(BX)=FG*BM(BX):BS(BX)=FG*AN:SH1(A*5+B)=0
2100      LINE(XX,YY)-(XX+4,YY+15),0,BF:PUT(BX*80,BB),BAR(0,BT(BX)),PSET
2110 BB=0:RETURN
2120 *BAR.MOVE
2130    BM(A)=BM(A)+1:LINE(A*80,BY(A))-(A*80+79,BY(A)+5),0,BF
2140    IF BS(A)=-1 AND BY(A)<25  THEN 2160
2150    IF BS(A)=1  AND BY(A)>369 THEN 2160 ELSE 2190
2160      PUT(A*80,0-(BS(A)=1)*395),BAR(0,BT(A))
2170      BC(A)=0:BY(A)=0:PN(BS(A)+1)=PN(BS(A)+1)+BT(A)+1:BS(A)=0
2180      BW=BW+1:IF BW=8 THEN GOTO *GAME.END ELSE RETURN
2190    BY(A)=BY(A)+BS(A)*2:PUT(A*80,BY(A)),BAR(0,BT(A))
2200    IF BM(A)=16*BC(A)*(BT(A)+1) THEN BM(A)=0:BS(A)=0:BC(A)=0
2210 RETURN
2220 DATA 27,10,"メロンヴァー・スイカヴァー"
2230 DATA 20,15,"Copyright(c) PUTDINK 2001 right reserved"
2240 DATA 33,18,"１Ｐ Com cs Man",33,19,"２Ｐ Man vs Man"
```



★デジカメ持って学校行ったら，先生に取り上げられた。下校時には返されたけど…。（香川県高松市・ネタなし15歳）……【つぐ美：もしかしたら，先生が使ってみたかったんじゃない？影：見たこともない画像が入ってなかった？ちゃんとチェックした？】

# Z-1GR，Z-1，FX-890P用



タイピングで２人対戦

# TipuraVer4.37

（ティプラ　バージョン）

間瀬　聖（18歳）

FX-890P

## 内容

画面の１行目に出てくる単語を打つタイピング・ゲームです。通信ケーブルを使って，２人対戦ができます。

## 遊びかた・操作方法

最初に，「Play(1 or 2)=」と表示されるので，１人か２人か選びます(２人で遊ぶ場合は,通信ケーブルでつないでく)。

１行目に「>」のつぎに単語が出てきます。それを打ち終わると，２行目に表示されているキャラクターが移動します。３行目には相手が表示され，１人ですするときは，自動で移動します。右のほうにあるラインまで先に移動したほうが勝ちになります。

慣れてきたら自動で動くスピードを速くするとよいでしょう。他の人とタイピング速度を競うと面白いと思います。

## プログラムの入力法と注意点

自動で動くスピードは，最初の行の「CXT=50」で，速くする場合は値を小さくしてください。

《第１表》変数表

| | |
|---|---|
| cf | 単語を読み込んでいるかどうか |
| mx | 自分のキャラクターのX座標 |
| yx | 相手のキャラクターのY座標 |
| cxt | 相手の動くまでのループ回数 |
| cxtk | 相手の動くまでのカウント |
| ke | 勝ち負けのフラグ |
| ng | つぎの文字の位置 |
| ngs | 単語の長さ |
| rn | 乱数で選んだ単語の位置 |
| i | ループのカウンタ |
| p | １人で遊ぶか２人で対戦か |
| kns | 単語の総数（0から255まで） |
| key $ | 押されたキーが入る（INKEY $） |
| kon $ () | 単語が全部入る |
| kont $ | 打ち終わった文字列（一単語分） |
| kan $ () | 勝ち負けの表示用 |

単語は，英字の小文字か大文字かどちらかに統一しておくとよいと思います。単語を増やした場合は，最初の行の「kns=74」を(単語の数-1)にしてclearを実行してください。単語は，255個までにしてください。

**注意**：２人対戦をする場合には，両方のポケコンにプログラムを入れ実行してください。

## 改造のポイント

・得点をつける。

・単語を英単語にする(今の単語は，ポケコンのBASICの命令)。

・カナが使えないので使えるようにする。

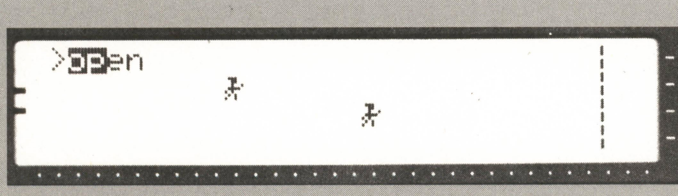
・２人対戦のときも自動で動くキャラクターを４行目につける。

・キャラクターの動きをコミカルする。

## CHECKER FLAG

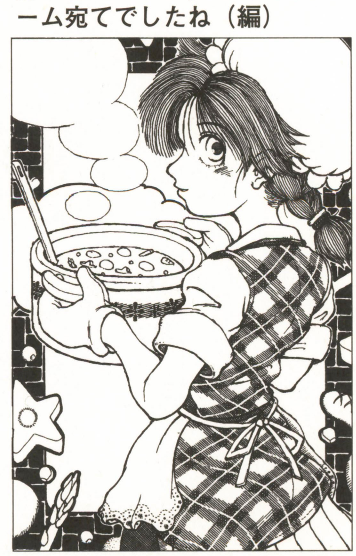
**編**：通信対戦がウリ言えますが，１人で遊んでも楽しめるゲームです。問題となる単語の書き換えが，リストから簡単に行えるため，自分が覚えたい英単語などにすると，勉強にも役に立つかもしれませんね？

**Dr.D**：単純なゲーム内容ながらも，対戦というアイデアを取り入れたため，遊べるゲームになったというわけじゃ。気になった点を言えば，タイピングの入力は，大文字と小文字どちらでも対応していると，遊びやすかったかもしれん。ちなみに，通信機能を使わない人は，この通信部分の処理を省けば，かなり短い記述でゲームが楽しめるゾ。




《写真１》単語を入力すると，キャラクターが移動していく

評…あっ！これ，あくせすルーム宛てでしたね（編）



by　わいち301

★OF掲載傾向と対策：「はじめてハガキを出します」と書くと必ず載る。しかも，コメントはつぐ美さんが答える場合が多い。どうですか，編集部のみなさん。（宮崎県日向市・**はじめてのハガキ17歳**）……【**つぐ美：必ずということもないと思うけどなあ。編：うーん。**】

『Tipura Ver.4.37』のプログラム・リスト

```
100 yx=1 :mx=1 :cxt=50 :cxtk=0 :ke=0 :ngs=0 :ng=0 :i=0 :kns=74 :p=1 :rn=0'C(2001) K,M
110 DIM kon$(kns),kan$(1) :kan$(0)="- You Win !! -" :kan$(1)="- You Lose !!-"
120 IF cf=0 THEN cf=1 :GOSUB 540
130 DEFCHR$(255)="054a7c6b0810" :CLS
140 INPUT "Play(1-2)=",p
150 CLS:LOCATE 4,1 :PRINT "- - tipura  Ver 4.37 - -";
160 IF p<1 OR p>2 OR p=0 THEN GOTO 140 ELSE IF p=2 THEN GOSUB 430
170 BEEP :WAIT 8 :CLS :POKE &H17B3,0
180 LOCATE 6,1 :PRINT "- start -" :WAIT 5 :BEEP :CLS
190 FOR i=0 TO 3 :LOCATE 29,i :PRINT "|"; :NEXT i
200 GOSUB 530
210 FOR mx=1 TO 29
220 LOCATE 0,0 :PRINT "             " :rn=INT(RAN#*kns+1)
230 ngs=LEN(kon$(rn) :ng=1 :kont$=""
240 LOCATE 1,0 :PRINT ">"+kon$(rn)
250 GOSUB 520
260 ON p GOTO 490,470
270 key$=INKEY$ :cxtk=cxtk+1
280 IF key$=MID$(kon$(rn),ng,1) THEN ng=ng+1 :kont$=kont$+key$ :BEEP ELSE GOTO 260
290 LOCATE 2,0 :PRINT REV kont$
300 IF ngs<>ng-1 THEN GOTO 260
310 IF p=2 THEN PRINT #1,mx
320 NEXT mx :GOSUB 520
330 IF mx>yx THEN ke=0 ELSE ke=1
340 BEEP :WAIT 8 :CLS
350 IF p=2 THEN PRINT #1,mx :CLOSE
360 LOCATE 8,1 :PRINT kan$(ke) :BEEP
370 LOCATE 9,2 :INPUT "Continue(y,n)",yn$
380 IF yn$="y" OR yn$="Y" THEN CLS :yx=1 :GOTO 150
390 IF yn$="n" OR yn$="N" THEN CLS ELSE GOTO 370
400 CLS :BEEP :LOCATE 10,1 :PRINT "- The End -"
410 END
420 '
430 OPEN "COM0:7,E,8,1,N,N,N,B,S" AS #1
440 TIMER=0
450 IF PEEK &H17B3>0 THEN PRINT #1,"*" :RETURN
460 IF TIMER>12 THEN PRINT #1,"*" :GOTO 440 ELSE GOTO 450
470 IF PEEK &H17B3>0 THEN INPUT #1,yx :GOSUB 530 ELSE GOTO 270
480 POKE &H17B3,0 :IF yx>29 THEN GOTO 330
490 IF cxt<=cxtk THEN yx=yx+1 :cxtk=0 :GOSUB 530
500 IF yx>29 THEN GOTO 330 ELSE GOTO 270
510 RETURN
520 LOCATE mx,1 :PRINT CHR$(255); :LOCATE mx-1,1 :PRINT " "; :RETURN
530 LOCATE yx,2 :PRINT CHR$(255); :LOCATE yx-1,2 :PRINT " "; :RETURN
540 FOR i=0 TO kns :READ kon$(i) :NEXT i :RETURN
550 '
560 DATA "list","edit","renum","new","delete"
570 DATA "run","cont","system","pass","save"
580 DATA "load","merge","bsave","bload","mon"
590 DATA "clear","defseg","mode","angle","set"
600 DATA "dim","erase","if","then","else"
610 DATA "for","to","next","goto","on"
620 DATA "gosub","return","input","cls","locate"
630 DATA "line","circle","paint","swap","print"
640 DATA "using","tab","draw","drawc","rev"
650 DATA "norm","open","close","data","read"
660 DATA "restore","write","rem","let","beep"
670 DATA "wait","stop","end","peek","poke"
680 DATA "tron","troff","varlist","error","resume"
690 DATA "format","files","chain","kill","name"
700 DATA "stat","call","outport","out","append"
```

**★参考にしたもの**

・スーパーカレッジ　Z-1GR　Software Library　C言語・BASIC・CASL編（カシオ計算機(株)制作／編集（株)モダン)

・スーパーカレッジ　Z-1GR Software Library　ガイドブック（カシオ計算機(株) 制作／編集（株)モダン)・

・I/O別冊　Z-1／Fx-890P活用研究(編集I/O編集部，発行所 工学社(株)1994年「P174～181，特にP178参照」)



# CMM 試聴室

ここで紹介しているデータは,CMMのWebSite (http://www.basicmagazine.net/cmm/) に収録されています。

## LAST SUMMER……自作

**投稿者:ナガタヒロカズ(23歳)**

**制作音源** MU90B

### かつてMSXで作った曲です

かつて高校生のとき,クラスの仲間とバンドを結成し,サザンのコピーとかをやっていたのですが,オリジナル曲を作ろうということになり,キーボード担当の私が曲を書くことになりました。当時私はMSX TurboRのFM音源で,曲を作っていましたが,音色が64種類しかないうえに最大同時発音数が10にも満たないという,非常に制約の強い音源(?)でした。この曲はそのとき作曲されたものです。バンドのリーダーが後から詞をのせる予定でしたが,いつまでたってもできあがらず,そのうちバンドも自然消滅してしまい,ついにこの曲が陽の目をみることはありませんでした。あれから8年,あらためてこの曲をふりかえってみると,やたら音域が広く,歌うのはかなり難しいことに気づきました。というわけで,この曲は歌モノというより,インストルメンタルとして聴いてください。

### 編集部から

メロディアスなポップス系の曲です。フュージョンと歌謡曲を足して2で割ったような感じでしょうか。構成がはっきりとしていて,メロディが非常に聴きやすい曲です。打ち込みやミックスも押さえるべき部分はしっかりと押さえられて安定感があります。

構成では,Cメロをソロで鳴らしてみたり,繰り返し部分にもう少し変化をつけたり,最後の繰り返しはキーを一つ上げてみたりと,もう少し工夫できると思います。1ループするまではいいのですが,その後,同じ調子でずっと鳴ってしまうので,途中に変化を与えたほうがいいでしょう。また,オーケストラヒットのフィルイン部分がぼやけ気味なので,音の芯の部分と余韻の部分を別の音で打ち込むか,リバーブのパラメータを調整して,原音をもう少しはっきりさせるとメリハリがついてさらに効果的になると思います。

## Holy Arrow………自作

**投稿者:小野寺秀善さん(21歳)**

E-Mail:BIBL0800@crest.ocn.ne.jp

**制作音源:** QY70

### 神秘的な感じを出してみました

今回のテーマは「聖なる矢」という神秘をテーマにしました。神のお裁きを受けるのじゃー!!

などという想像をしてしまいました。あまり関係ないかも……。そんなことはおいておいて,かなり打ち込みには時間がかかりました。イントロだけでかなり時間がかかりました。

曲の内容なのですが,曲はデジタルなクラシック風?とでもいいましょうか?メロディ・ラインがかなりシンプルになってます。古代のローマ帝国のような感じもしますが,Hichriki(能楽器)をメロディ楽器に持ってきたのが良かったのかと,それと14トラックでアナログボイスを高音帯域で鳴らして,また途中転調しますが,また元の調に戻ります。これでかなり神の神秘が再現できたと思いますが,ちょっと怖い感じになったかな?神の神秘を感じ取ってください。

### 編集部から

各音の打ち込みに非常に凝ったデータですね。曲的には,シンプルで聴きやすいところもいいと思います。最近こういった,音で遊ぶ曲の投稿が少ないので,久々に新鮮な感覚で聴くことができました。

その音についてですが,全体的に腰がすわっていない印象を受けます。すべての音に深いリバーブがかかっていて,中心の楽器の定位感が薄れてしまっているのが原因だと思うので,たとえばベースやバスドラムといった低音を担当する楽器は,ある程度ドライにするか,上のコメントでも書きましたが,芯の部分とエフェクトの部分を分けてみるといいと思います。また曲自体ですが,ちょっととりとめのない,まとまりのない感じもあるので,聴かせたい音とともに,聴かせたいフレーズも全面に出してみるといいでしょう。

### ベーマガホームページで曲が聴ける!!

掲載された曲に関しては,規定の原稿料をお支払いします(雑誌発売後1か月程度かかります)。曲データは,CMMのWebSite(http://www.basicmagazine.net/cmm/)にもコメントとともにアップロードされます。ぜひ聴いてください。



●応募方法:本誌とじ込み愛読者ハガキ(またはインターネット読者ハガキ)にプレゼント記入欄があります。一つだけ選んでお送りください。●締め切り:12月7日必着《10月号の当選者》①大阪府吹田市・酒井 巌さん他4名 ②滋賀県守山市・西山竜一さん他1名 ③熊本県菊池郡・川辺裕喜さん

マイコンBASICマガジン

# 明日のスター☆プログラマー

## 今月のベスト・プログラマー

**編：**2001年も12月号。毎年，この時期になると，1年立つのは早いものだな，と実感しています。今年1年間，投稿プログラム・コーナーに参加してくださったみなさん，ならびに応援してくださったみなさん，本当に…。

**編集部一同：**ありがとうございました。

**影：**さて，12月号といえば恒例の「2001年スター・プログラマー」の発表だ。

**Dr.D：**うむ。じゃが，その前に忘れてはならんのが，今月のベストの発表だな。今月はVBの『WIND』，VBScript＋DirectXの『Touch Down』の2作品が好評だった。

**つぐ美：**リスト，ゲーム内容が良かった『WIND』，トランスフォームからフライトゲームを作るというアイデアが評価された『Touch Down』ってとこでしたね。

**影：**編集部でも，意見が分かれたけど，今月は僅差ながら『Touch Down』をベストに選びました。




▲VBScript＋DirectX用『Touch Down』

## 年間スター・プログラマー発表

**Dr.D：**それでは，いよいよ「2001年スター・プログラマー」の発表じゃな。まずは，今年1年間のベスト・プログラムを振り返るとしよう。

| 掲載号 | 投稿者 | 作品名 |
|---|---|---|
| 1月号 | 浜田明巳 | 五心シミュレーション |
| 2月号 | 関谷昌平 | ゲートリミット |
| 3月号 | 吉田真人 | 習字道場 |
| 4月号 | 大窪　諭 | げっとパズル |
| 5月号 | 中尾孔一 | GTool |
| 6月号 | 山下陽介 | 株式皮算用 |
| 7月号 | 谷澤友哉 | 岩の開拓者 |
| 8月号 | MONOMON | マウスde世界陸上 |
| 9月号 | 吉田真人 | Word Hunter |
| 10月号 | ヅラChu | 牛丼食えスト外伝　牛丼道 |
| 11月号 | INF | 走り跳び逃げ |
| 12月号 | 金井義男 | Touch Down |

**つぐ美：**こうしてみると，今年もいろいろな作品がありましたよね。

**影：**さて，検討の結果，ついに決定した「2001年スター・プログラマー」は…？

**Dr.D：**『習字道場』，『WordHunter』とVBでアイデアに富んだ作品で2度のベストに輝いた吉田真人さんに決定じゃ。

**編：**おめでとうございます。賞品として「ＸＲＧＢ-2」を贈呈いたします。誌面からではありますが，編集部全員から大きな拍手を。

**編集部一同：**パチパチパチパチパチ。

**つぐ美：**来年も引き続き，みなさんからの投稿プログラムを心よりお待ちしています。来年もよろしくね。

## 投稿ありがとう

Shota／高橋好人／間瀬　聖／阿部雅之／M.Hayashi／ジェイエム／りっちもんど／ていこく／金井義男／仁井内明／怪盗プー／ほおく／二宮裕志／石山明弘／沖田慎吾／大内慶裕／橘　一馬／プップリン／canopy／みみずく／藤本　豊／久保田尚彦／SPP／黒鵜嘉久／バビブベボーちゃん改めバビブ／高橋恵一／東海恭助／森　博行／NAOKI／平原充広／平林　豪

※全員紹介できなくてごめんなさい

## 今月の掲載者から一言

**Windows (Visual Basic V6.0)用**
**「目指せ!!千切りマスター」**
**高橋好人（18歳）**

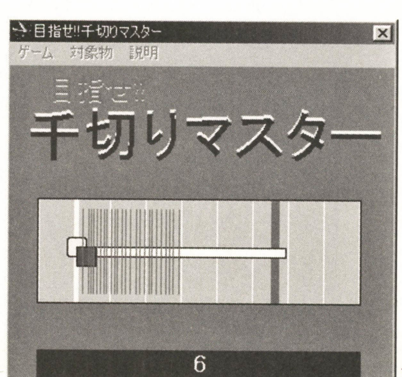
**作成後記**

なんか知らないけど，このゲームにハマってしまった。マウスのボタン連打ゲームは結構あるけど，いろんな要素を加えたし，何より千切りをゲームにしたのは，オレ以外にいないと思う。だけどちょっと，サウンドとかに凝ったのでリストが長い。VBってQBやN88に比べてかなり長くなりがちだね。

ところでどうでしょう？高得点が取れましたか？このゲーム（実際の千切りでも）正確さは，すべての対象物（食材）に共通することがわかっていただけると幸いです。もし感想があれば，tayosan@tokutoku.or.jpまで送ってください!!

あと，私のハイスコアをお知らせしておきます。MAXというのは，取れ得る最高得点です（あっているかは自信なし）。すべてのモードでハイスコアを狙ってください！

・長葱：896点，MAX：1300点
・大根：1712点，MAX：3220点
・ピーマン：353点，MAX：444点






▲HSP用『走り跳び逃げ』

## 投稿プログラム・コーナー

# Programmer's Room

### 今月の総評

**編：**2001年の締めとなる12月号。今月は，個性のある作品が多く，アイデア重視のベーマガらしい投稿状況だったと思います。

また，今月から超大作作品に新たな試みを行いました。もうお気づきかと思いますが，その試みとは「Pick Up List」というコーナー。このコーナーは，すべてのリストを掲載できない超大作作品のリストの一部分を抜粋し，そのリストについてDr.Dにコメントしてもらう，というものです。リストの掲載をウリにしているベーマガらしさを，超大作へも反映させた形になりました。ぜひ，この試みに対するご意見を，愛読者ハガキなどでおよせください。

### 2001年の総評

2001年のベーマガは，4月号からプログラミング重視へと生まれ変わりました。そんな中，「投稿プログラム・コーナー」は，この1年間で，CHECKER FLAGのコメント増加，Programmer's Room，Dr.Dのベスト・プログラム研究室の立ち上げなど，新しい一歩を踏み出せたと思います。

投稿面では，iアプリをはじめ，JavaやWonderWitch，そしてUNIXなどバラエティに富んだ言語の作品の投稿があり，新しい投稿の流れが生まれました。

また，投稿者では，中学生プログラマーからの投稿が多く，新しい読者からの投稿が期待できる状況になってきました。

来年以降も，読者に関心を持ってもらえるような「投稿プログラム・コーナー」にしていきたいと思っています。

「投稿プログラム・コーナー」は，読者の参加があってはじめて成り立つコーナー。ぜひ来年以降も，みなさまの積極的な投稿をお待ちしています。

### 一口CHECKER FLAG

**●Visual Basic用『あーる・かるきゅれーしょん』／ていこく(22歳)**

**つぐ美：**三つの変数の値を暗記して，その変数を利用した四則演算の問題を解答するゲーム。単純なアイデアながらも楽しめるゲームでしたが，チョット簡単すぎるかな，という印象です。問題の式の形が，もう少しバラエティに富んでいるといいんじゃないかしら？

**●Visual Basic用『Sf Synchronizer』／ほおく(26歳)**

**影：**指定したフォルダのバックアップをしてくれるソフト。細かな指定ができるけど，基本的にはコピー動作だけなので，実用性に欠ける印象だった。

**●Visual Basic用『VB to 旧Basic』／高橋好人(17歳)**

**編：**QBasicのプログラムをVBへコンバートするソフト。目の付けどころはいいのですが，肝心のコンバート作業を手作業で行うのが面倒ですね。この作業をある程度でもソフト側で行ってくれると実用性がグッと向上してくるハズです。

**●Visual Basic用『究極音感』／Shota(14歳)**

**Dr.D：**タイトルどおり，ソフトから再生される音を答えるというゲーム。プログラム面ではしっかりできているのじゃが，普通の人では，一番簡単なモードでさえ，クリアできないじゃろう。遊びやすいゲームを作る，ということも考えてゲームを作ってみてはどうかな？他の人にプレイしてもらって，助言してもらうのがオススメじゃ。

**●Visual Basic+DirectX用『怪鳥ミュー』／M.Hayashi (51歳)**

**Dr.D：**ブロック崩しに，シューティングやクイズといったさまざまな要素を取り入れたゲーム。ゲーム・アイデアとしては独創的で評価できるが，ゲームとしての質がイマイチと言わざるを得ない。肝心のボールの跳ね返り角度が45度固定だったり，バーとボールとの当たり判定もイマイチ安定していなかったりしている。ゲームとしては，しっかり形になっていただけに，余計に前述の指摘部分が気になったというわけじゃ。ボールの動きや当たり判定などの，ゲーム自体のクオリティを向上させる努力をしてみてはどうかな。

**●FX-890P用『Quick Timer』／八木克彦(34歳)**

**つぐ美：**タイトルから連想できるとおり，タイマ機能を持つプログラム。やはり基本的な機能なので，物足りなさを感じてしまいました。

**●N88互換BASIC for Windows用『Translation Composer ver.1』／石山明弘(33歳)**

**つぐ美：**11月号掲載のQBasic作品の移植版。ほとんど同じ形のため，新鮮味に欠けました。また，投稿面での注意ですが，メール投稿の場合，リストのファイルなどはLHAで圧縮し，それを添付したメールを1通だけ送信してきてください。持っているメール・アドレス全部から，同じ物がたくさん送られてきても困るので。あと，住所などの個人情報も投稿ごとに説明ファイルに記入してください。よろしくお願いしますね。

### 今月はもう一歩でした

**●VBScript+DirectX用『万華鏡』／金井義男(39歳)**

**影：**DirectAnimationの特徴をしっかり打ち出している作品。演出効果もバッチリでなかなかの出来なんだけど，動作速度やリストの長さが気になった。金井さんは，DirectAnimationの投稿作品では，もはや常連なので，これからは，ベーマガへの投稿を意識した作りをしてほしいなぁ。作品の完成度もさることながら，リスト面でのこだわりも見せてほしい。

# Dr.Dの LABORATORY ベスト・プログラム研究室



## 今月のベスト・プログラム：VBScript＋DirectX用『Touch Down』(68ページ掲載)

今月のベスト・プログラムは，VBScript＋DirectX用としては初となる『Touch Down』じゃ。本来ゲーム制作がむずかしいと思われるDirectAnimationを使用して，擬似的ながらも3Dフライト・ゲームを作ったのが評価された作品。今回は，このゲームのメインとなるトランスフォーム部分と，キー入力などを含めたゲームを構成している部分を解説してみよう。

(Dr.D)

## トランスフォーム処理

この作品で使用されているトランスフォームは，HeightFieldトランスフォームというものじゃ。この処理については，ここでも簡単な解説をするが，本誌2000年8月号のDirectX講座で，よりくわしく取り上げておるので，興味のある読者は，ぜひそちらにも目を通してほしい。

このHeightFieldトランスフォームは，画像ファイルの明るさを高低差に変換し，3D形式のようにするトランスフォームじゃ。この作品では，“picture.jpg”の画像に対してトランスフォームを適用しておる。また，山の大きさもトランスフォームに与える値によって変更可能じゃ。これらを利用した改造を，後の項目で紹介しよう。

このトランスフォームを使用することで，山のような地形が簡単に再現できるというわけじゃ。ただ，作り出した山に対して，当たり判定を付けようとするのは，むずかしいじゃろう。実際，この作品も山に対しての当たり判定は行っていない。

## ゲーム構成

では，トランスフォームを適用した画面を用いて，どのようにゲームを構成しているのかを見ていこう。

《リスト1》スクロール処理部分のリスト

```
'ジオメトリの回転と移動
 Set geo = geo.Transform(m.Rotate3Anim(m.ZVector3, ms))
 ※この命令で常にスクロールを行う。移動量となるmsは，
   NextTimeルーチン内の，「ms.SwitchToNumber (ms.Extract - 0.001)」
   という命令で常に変化している。

 Set geo = geo.Transform(m.Rotate3Degrees(m.XVector3, -80))
 Set geo = geo.Transform(m.Translate3(0, -0.4, 0))
 ※上記二つの命令で，適切な表示がされるよう，
   トランスフォームを適用した画像を移動させる
```

まずは，飛行部分(スクロール)について。スクロール処理を行う部分は，**リスト1**にコメントを入れて掲載してある。スクロールを行うのは，一つの命令だけで，あとの二つは3D化された画像が正しい位置で表示されるように調整するための命令と言える。また，スクロールを行うために，NextTimeルーチンでタイマーカウンタを実行し，スクロールに使用する値(ms)を時間に合わせて変更しておる。

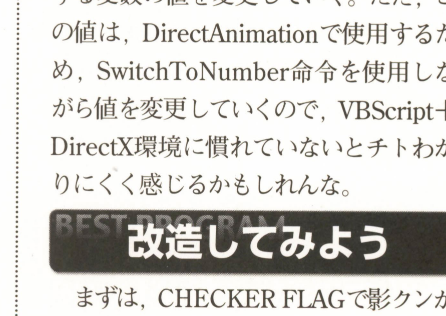
つぎに着陸部分の設定と，着陸判定について。着陸判定は，Checkルーチンで行っておる。判定方法は，if文の条件式内で，mx（バランスの値の判定），mz（高度の判定），ms（ゴール地点の場所であるかを判定）の値が，着陸の条件を満たしているかを判定し，着陸していればエンディングのイメージへ切り替え，着陸失敗ならばバランスや高度などを再設定し，引き続きゲームを続行させる。

msの判定用の値が固定されていることからわかると思うが，着陸地点は常に固定された場所になる。また，着陸部分画像を表示する座標も別に設定されているため，着陸地点を変更するのがむずかしくなっておる。

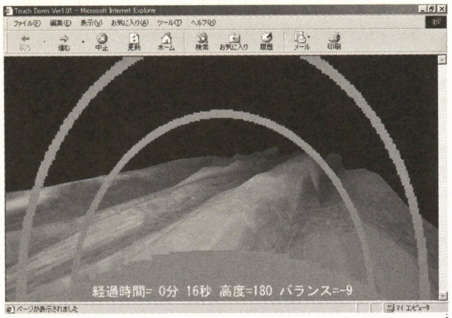
最後にキー入力について。キー入力は，DAControl_KeyDownルーチンで処理をしておる。キーの区別は，KeyCodeに各キーに対応した値が入るので，Select～Case文で値に対応した処理を行う。ゲーム中で使用するカーソル・キーなどの入力があった場合，動作に関連する変数の値を変更していく。ただ，この値は，DirectAnimationで使用するため，SwitchToNumber命令を使用しながら値を変更していくので，VBScript＋DirectX環境に慣れていないとチトわかりにくく感じるかもしれんな。

## 改造してみよう

まずは，CHECKER FLAGで影クンが触れておるキー入力の変更について触れよう。




《写真1》適用画像をdx.bmpへ変更した場合の実行画面



《写真2》リスト2のプロパティ改造例の実行画面。平坦で横長な山が作られる

キー入力については，前述で説明したとおり，KeyCodeで区別しておる。ということは，このKeyCodeに対応させるCase文の値を変更すれば，キー対応も変更されるというわけじゃ。

たとえば，↑↓キー入力の処理を入れ替えるには，Case 38とCase 40の値を入れ替えればOKじゃ。

つぎの改造は，HeightFieldトランスフォーム処理を変更した改造。このトランスフォームは，画像ファイルに適用されるため，適用する画像ファイルを変更すれば，その画像に合わせた地形が作り出されるわけじゃ。**写真1**は，適用画像をdx.bmpへ変更した場合の実行画面じゃ。少しわかりにくいかもしれんが，地形が変化しているのを確認してほしい。自分で実行すれば一目瞭然じゃゾ。

HeightFieldトランスフォームは，設定されているプロパティの値を変更することで，山の作りかたを変更することができる。このプロパティについては，前述の本誌2000年8月号掲載のリファレンスが役に立つじゃろう。**リスト2**に，説明コメントを加えた変更リスト例を掲載してみたので，このリストを参考にして，HeightFieldトランスフォームのいろいろな動きを試してみてほしい。

最後にワシがCHECKER FLAGで触れたゲーム性の面での改造を行ってみよう。何回も述べているがVbscript＋DirectXの環境では，変数の扱いが面倒になるなどで，ゲーム制作はむずかしい。そのため大幅な改造はせず，飛行機の動きを変更するなどの改造で，より遊べるようにしてみよう。簡単なところでは，左右のバランス機能を生かす改造じゃな。**リスト3**に改造リストを掲載してみた。とはいっても行っているのは，↑↓キーを押すごとにmxの値を加算，もしくは減算しているだけ。じゃが，この記述をすることで，バランス調整を考えながら高度を上げ下げするするようになるため，よりゲーム性が増すわけじゃ。飛行機の動きとしては，多少不自然になるが，このような簡単な記述でもゲームの印象を変えられるということは覚えてもらいたいのぉ。

《リスト2》プロパティ設定部分のリストならびに改造リスト

```
'DirectX Transformプロパティの設定
  dxt.Depth = 1
  ※盛り上がる部分の最大量。マイナスの値を設定すると，谷が作られる。

  dxt.Width = 4
  dxt.Height = 4
  ※生成される山の幅，ならびに高さを設定する。

  dxt.Samples = 65
  ※この作品では記述されていないが，Samplesというプロパティも
  あり，画像の縦横の分割数を示す。この値が大きいと，より詳細な
  3D化を行うが，その分ポリゴン数も多くなり，処理も重たくなる。
--------
プロパティ改造例
  dxt.Depth = 1
  dxt.Width = 20
  dxt.Height = 4
  dxt.Samples = 20
  と設定する。平坦で横長な山が作られるようになる。
  《写真2》が実行画面。
```

《リスト3》キー操作部分の改造リスト

```
Case 38  '[↑]:高度を上げる（スピードアップ）
    If flag = False Then
      mx.SwitchToNumber (mx.Extract - 0.01)  ※追加部分
      ms.SwitchToNumber (ms.Extract - 0.02)
      mz.SwitchToNumber (mz.Extract + 0.05)
（以下同じため省略）

Case 40  '[↓]:高度を下げる（スピードダウン）
    If flag = False Then
      mx.SwitchToNumber (mx.Extract + 0.01)  ※追加部分
      mz.SwitchToNumber (mz.Extract - downval)
（以下同じため省略）
```

## で・ば・ぐ

2001年11月号掲載のVB用『普通のタイピング・ゲーム』で，Project2のフォーム，プロパティ表にDSラベルの記述が抜けていました。

DSラベルはTSラベルの上に重なる位置に貼り付けます。プロパティは以下のとおりです。

申し訳ありませんでした。（編集部）

DSラベルのプロパティ

| | |
|---|---|
| Name | DS |
| BackColor | &H00C0FFC0& |
| Height | 255 |
| Left | 120 |
| TabIndex | 4 |
| Top | 1560 |
| Width | 4455 |

# PROGRAMMING GET&RUN（ゲット＆ラン）

プログラムの疑問・質問にお答えします

**アルゴリズム全般**

**言語基礎知識**

**Q** はじめまして，本年度大学受験に合格し，自分専用のパソコンを買いました。

一番上の兄は，以前からベーマガの愛読者で（1986年前後から），最初の愛機は『PC-6001mkⅡSR』だそうです。当時BASICを学び，その後マシン語ができるというステータスだったそうです。

さて，兄に紹介されて，ベーマガのホームページに行き，プログラムをダウンロードさせてもらいました。入学祝として，『Microsoft Visual Basic（以下VBと略）6.0』を入手したので，これからいろいろトライしたいと思っています。

質問があります。

兄の話ですと，「コンピュータを本当に理解したければ，マシン語を勉強するといい」とのことです。「マシン語がわかれば，コンピュータは怖くない。ただの計算の速い大きい計算機」といっていました。

現在は，どのような環境でマシン語を勉強できるのでしょうか。ぜひ，教えてください。

また，“C言語”や“C＋＋”は，『Microsoft Visual C＋＋（以下VC＋＋と略）』とはちがうとのことですが，どうちがうのでしょうか？

本誌を読んでいたら，『ワンダースワン』がC言語を使っているということですよね。それについても教えてください（いずれVC＋＋も入手予定）。

はじめてのことなので，質問ばかりですが，どうかよろしくお願いします。

【？・田中佑治？歳】

**A** たくさん質問があるので，一つずつ回答していきましょう。

## マシン語とは何か

1980年代に主流だった8ビット・パソコンでは，機能や開発環境などによる制限から，プログラミングは「BASICかマシン語か」という2択を強いられた場合が多くありました（マシン語は機械語ともいわれる）。一般的にはBASICでプログラミングするのですが，速度的，機能的に不可能なことはマシン語を使うというのが当時の時流だったのです。

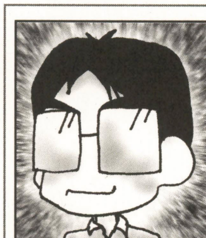
それに対し，現在ではC言語などの高級言語（人間が理解しやすい言語）が利用しやすく，マシン語にくらべて速度的にもそれほど遜色ない状況となっています。開発の効率や汎用性などから，昔はマシン語を用いていたような分野でも，一部をのぞいてC言語など高級言語が使われているというのが実情でしょう。

そのため，現在ではマシン語を学ぶための資料などの入手がしにくく，また習得しても，実際にプログラムする場が減りつつあります。

ですが，田中さんのお兄さんがいうように，コンピュータそのものを学ぶのには，マシン語の習得は非常に役に立つと思います。マシン語という，ハードウエアと密接に関係した環境でのプログラミングを経験することは，コンピュータそのものの構造，動作の理解を助けます。

たとえ実際にマシン語でのプログラミングを必要とされなくても，マシン語を学ぶことで身につけたコンピュータの知識は，いろいろな面で役に立つことでしょう。

TEXT&PROGRAMMING●山崎 寛



「M5」や「MSX」でプログラミングを覚えた，BASIC歴15年の男。現在，iPAQ（コンパックのPocket PC）を使っているけれど，手書き文字認識やソフト・キーボードでの入力はやっぱりめんどうで，立ったまま使える小型のキーボードがほしい。どうやら，10月にアメリカで発表された新iPAQでは，小型キーボードが用意されるようだ。ちょっと気になるところ。

## 今ならどんな環境がある？

つぎに，マシン語を勉強する環境を紹介しましょう。

マシン語そのものは，現在Windowsなどで一般的に使われているパソコン（PC／AT互換機）でも動かすことは可能です。ただ，前述したようにマシン語の必要性が低い現在，満足のいく資料や解説を入手するのはむずかしいと思います。そのため，「マシン語を習得する」ということは若干むずかしくなることは覚悟し

### 疑問・質問ないですか？

このコーナーでは，プログラムに関するQ＆Aを掲載しています。

プログラムに関する疑問（「こんなゲームを作りたいんだけど，ここがうまくいかない！」とか「このプログラムの入力方法がわからない〜」）などがありましたら，なるべく具体的に，使用機種や言語を明記して，下記住所までハガキや封書もしくは電子メールでお送りください。住所，氏名（ペンネーム），年齢，職業，電話番号も忘れずに。

■郵便あて先
〒141-8715　東京都品川区東五反田1-11-15
電波新聞社マイコンBASICマガジン編集部
GET＆RUN係

■電子メールあて先
bm@jolls.co.jp

※メールのタイトル／Subjectに「GET＆RUN係」と明記してお送りください。

《第1表》プログラミング言語の種類と製品

| 言語名（読み） | 製品名 | メーカー |
|---|---|---|
| BASIC（ベーシック） | Microsoft Visual Basic | マイクロソフト |
| C++（シープラスプラス） | Microsoft Visaul C++ | マイクロソフト |
| PASCAL（パスカル） | Borland Delphi | ボーランド |
| PASCAL（パスカル） | Borland Kylix | ボーランド |
| C++（シープラスプラス） | Borland C++Builder | ボーランド |
| Java（ジャバ） | Borland JBuilder | ボーランド |
| BASIC（ベーシック） | F-BASIC | 富士通 |
| C（シー） | ワンダーウィッチ（Cコンパイラを同梱） | キュート |
| BASIC（ベーシック） | BASIC STUDIOパワフルゲーム工房 | アートディンク |

ておいたほうがいいでしょう。

マシン語を学ぶ場合，実際にはマシン語と一対一に対応した“アセンブリ言語”という言語を扱います。そのため，アセンブリ言語からマシン語への変換をする“アセンブラ”というソフトを使うのが一般的です。

アセンブラには，いろいろな種類があります。有名なものとしては，マイクロソフトの『MASM』やボーランドの『TASM』などがありましたが，かなり古いソフトなので現在は入手がむずかしいでしょう。

それよりは，同等の機能を持った他のソフトを使うほうがいいかもしれません。とくに海外では，フリーのオンライン・ソフトとして，アセンブラが多く公開されています（また，C言語のコンパイラなどで，インライン・アセンブラとして，直接アセンブリ言語を扱うことのできる環境もある）。

ただし，現在のパソコンはハードウエアもOSも，昔の8ビット・パソコンからくらべると非常に複雑なものになっています。個人的な意見としては，コンピュータそのものの構造を勉強をするのには向いていないかもしれません。構造がシンプルなコンピュータ（ワンボード・マイコンや昔のパソコンなど）のほうが，機能は限られますが，むしろ全体を把握しやすく，わかりやすいのではないかと思うのです。

そういった意味では，質問にも登場したワンダースワン（ワンダーウィッチ）などは，比較的シンプルなハードウエアです。コンピュータの勉強向きといえるかもしれません。

## C++とVC++のちがいは？

つぎに，C++とVC++のちがいについて回答しましょう。このちがいは，C++は言語仕様のことで，VC++（Microsoft Visual C++）は製品名ということになります。

C++は，BASICやPASCALといった，言葉の種類のことを指します。VC++も，言語としてC++を採用していますが，他にもC++を採用した製品はたくさんあります。これら製品では，C++の言語仕様にある部分については同じようにプログラムを記述することができ，互換性があると思っていいでしょう。

ただし，C++の言語仕様というのは，プログラムを記述する基本的な部分についてだけで，その他については製品ごとにちがうライブラリ（共通で使える部品）が用意されていたりと，互換性のない部分があります。そのため，実際のプログラムでは，プラットフォーム（OSなど）に依存する部分は互換性がなくなる場合が多くなります。

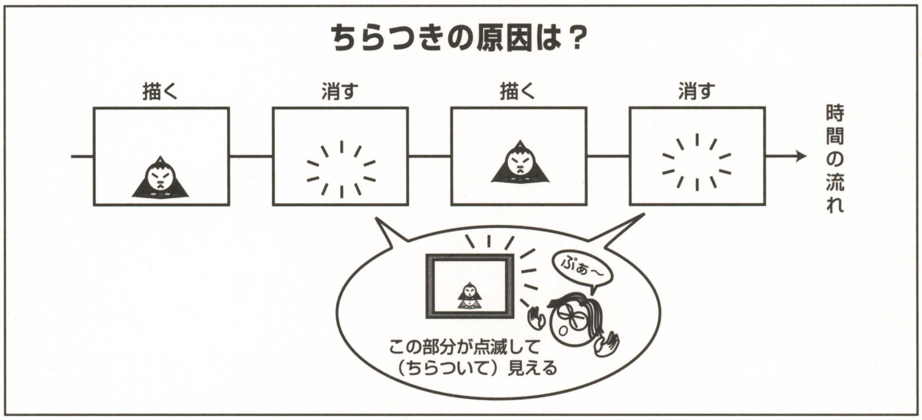
採用しているプログラミング言語の種類と，製品の種類などを，簡単にまとめてみました（**第1表**）。このように，同じ言語を採用している製品はいくつかありますが，これらはまったく同じプログラムを利用できるわけではありませんので注意してください。

## アルゴリズム全般

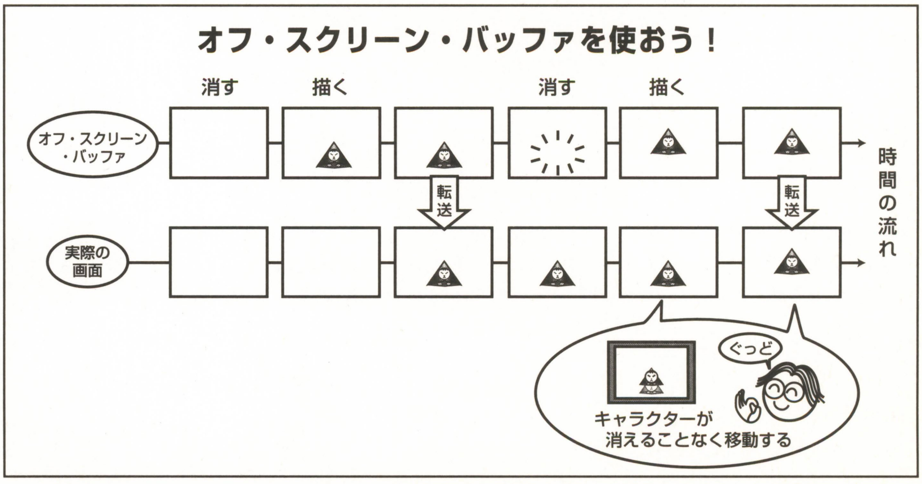
**Q** 突然の質問で失礼いたします。毎月この雑誌を読ませていただき，大変勉強になっております。

質問ですが，今VC++で，シューティング・ゲームを作っています（正確にはVBで一回作ったものを移植している）。

その際，自機，自機弾，敵キャラクタ




《第1図》ちらつきの原因は，画面でキャラクター消去をすることなのだ




《第2図》ちらつきをなくすにはいろいろな方法があるが，このオフ・スクリーン・バッファが一般的だ

ーなどを，ピクチャボックスに直接描画しているせいか，とてもちらつきます。そこで，一回どこかに描画してから，ピクチャボックスに描画したいのですが，その方法がわかりません。

一回どこかに描画して転送するやりかたでも，その他の（ちらつきを抑える）方法でもどちらでもかまいませんので，教えてください。よろしくお願いします。

**【東京都狛江市・ウイング27歳】**

**A** ちらつきに関する質問ということで，まずは環境に関係なく，なぜちらつきが発生するかということを考えてみることにしましょう。また，ウイングさんはある程度プログラミングができるという前提で回答していきます。

## ちらつきの正体は？

まず，“ちらつき”とは何でしょう？

通常，ゲームなどを作る場合，自機や弾，敵などのキャラクターを動かすため，描いては消し，つぎの場所に描いては消す，ということを繰り返しますよね。

これを図にすると，**第1図**のようになります。この「消した状態」が画面に現れてしまう（見えてしまう）と，キャラクターが出たり消えたりして見えます。

これが，“ちらつき”になるわけですね。

## オフ・スクリーン・バッファを使う！

ちらつきをなくすためには，キャラクターを消した状態を表示しない必要があります。

そのための一つの手法として，キャラクターなどの描画は画面に直接行わず，オフ・スクリーン・バッファに行う方法があります。オフ・スクリーン・バッファ（表示をfalseにしたピクチャーボックスなど）に対してすべて描画した後に，バッファの全体を画面に転送することで，ちらつきがなくなります（**第2図**）。

この方法を使う場合，直接画面に描画するのにくらべ，最後にバッファから画面に転送するという工程が加わるため，描画速度は落ちます。そのため，表示画面と同じくらいの大きさを，バッファから画面に高速転送できなければなりません。

WindowsのGDIでプログラムするならば，このような場合Windows APIのBitBltなどを利用するのがいいでしょう。

BitBltは，ピクチャボックスの一部分を，他のピクチャボックスなどの指定した部分に転送することができます。ですから，オフ・スクリーン・バッファに対してキャラクターなどを描画する代わりに，あらかじめ描いておいた内容をピクチャボックスなどから転送する場合にも使えます。

また，DirectX（ダイレクトエックス）を使って描画している場合は，“フリッピング”と呼ばれる，裏画面と表画面を入れ替える機能を使うといいでしょう。GDIにくらべて，高速にちらつきのない表示が実現可能となります。

それでは，また来月。




前回私が口座を作ったジャパンネット銀行はYahoo!オークションのオフィシャルバンクということで，思いきってオークションに挑戦！　Yahoo!オークションの本人確認登録は簡単にネット銀行で済ませられました。

さてまずは落札してみよう！　ってことで欲しいモノをリサーチ。もう売っていない･･･でも手に入れたいレアモノを手に入れるにはうってつけの場所だと思った私が目をつけたのはたまごっちの指人形。今時なかなか欲しい人いないだろう･･･と思っていたら，一人すごい強敵現れ争奪戦に！　後でプロフィールを見てみると，指人形マニアの人だったんですが，けっきょく私が落札。でも夢中になっていたら値段が100円から1500円にも跳ね上がってしまって駆け引きが難しい！　と思いました。

（神奈川）

Yahoo!オークション
http://auctions.yahoo.co.jp



パソコンでロボットを操る！

# レゴ道場 桐林部屋

# チャレンジ!! レゴ・マインドストーム

LEGO MINDSTORMS ROBOTICS INVENTION SYSTEM




## お題：センサのちょっと進んだ使いかた

こんにちは，桐林です。今回は少し進んだセンサの使いかたを紹介します。変わったライントレーサの作りかたもあるので，ぜひ作ってみてください!!（文・写真／桐林星河）




桐林星河（きりばやしせいが）
機械やロボットが好きな14歳。小学生の頃からメカに慣れ親しみ，LEGO MINDSTORMSの競技などでは常に好成績。TV出演なども多数。現在中学3年の受験生。

### RCXのセンサポート

RCXにはセンサポートが3か所あります。しかし，大きいマシンや複雑なマシンを作るとセンサの数のほうが多くなってしまい，困ってしまう場合があります。

そういうときは「センサの並列つなぎ」です。並列というと，電圧がどうのこうのとか，電流がどうのこうのというのが頭に浮かび，なんだか難しそうですね。しかし，レゴの場合はそんなに難しいものではありません。ただ単に，センサやモータのケーブルを縦に重ねて，複数の負荷に電気を行きわたらせるだけなのです。

### つなぎかた

実際につなぐと，こういう感じになります。モータの場合は，重ねたケーブルのモータ同士は同じ動きをします。だから，相撲ロボットなどでモータのパワーを上げたいときはこの方法を使います。ただ一つ，注意しなければいけないのが，つなぎかたです。まちがっても，同じケーブルの端と端を重ねてRCXや電池ボックスに付けないように気をつけてください。電流がケーブルの中だけで流れてしまってとても危険です（**写真1，2，3**）。

センサを重ねる場合は，センサの種類が関係してきます。では，いろいろなパターンを紹介しましょう。

### タッチセンサ

タッチセンサ2個だと，BASICでいうor文にあたります。くわしくは**第1表**を見てください。押された状態が1，離された状態が0で表されています。

タッチセンサの場合は，2個以上つけても平気です。これを利用して，センサポートを節約したり，確実性を増したり，いろいろとできます（**写真4，5**）。

**《第1表》タッチセンサ2個つないだときの結果**

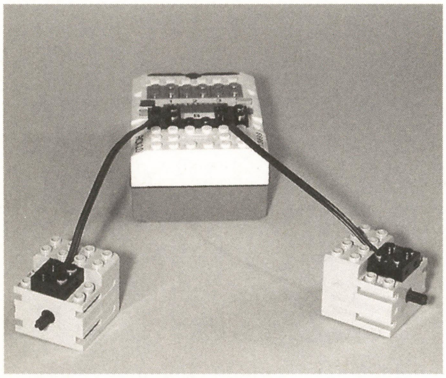
| パターン | 入力1 | 入力2 | 結果 |
|---|---|---|---|
| パターン1 | 0 | 0 | 0 |
| パターン2 | 1 | 0 | 1 |
| パターン3 | 0 | 1 | 1 |
| パターン4 | 1 | 1 | 1 |

※1＝押された状態，0＝離された状態

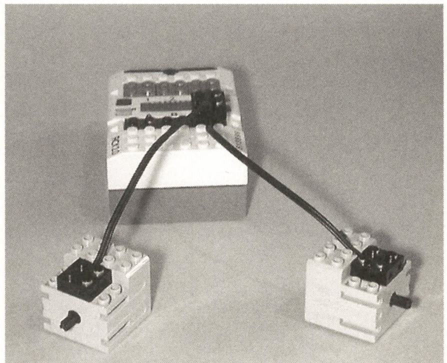
### ライトセンサ

ライトセンサは0～100の値で光の明暗を見分けています。0がもっとも暗く，100に近づくほど明るい状態を示しています。これは，明るさのちがいが微妙だと，働かなくなってしまう危険があります。そこで今回の技を使います。

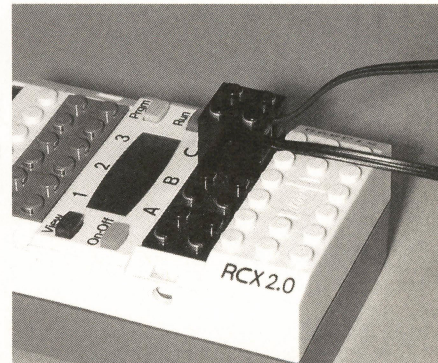
ライトセンサ2個をつなげた場合，値の差が大きくなります。たとえばライトセンサ1個の場合，暗いときが35，明る



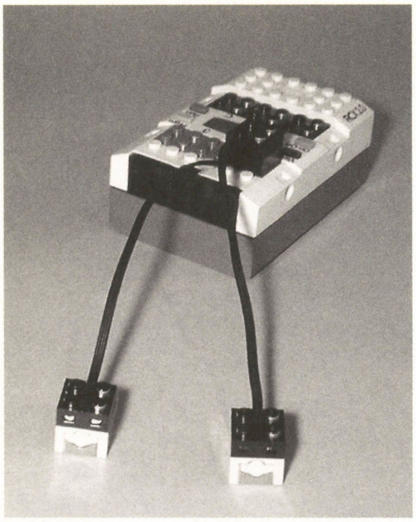
《写真1》普通にRCXにモータを2個つけた状態



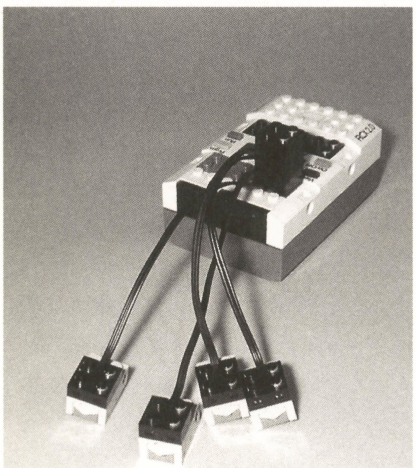
《写真2》同じポートにつけた状態。この状態だと同じ動きしかできない




《写真3》ポートのアップ



**《写真4》タッチセンサを2個つなげた状態**



**《写真5》4個つけても問題なし。何個つけても平気です**

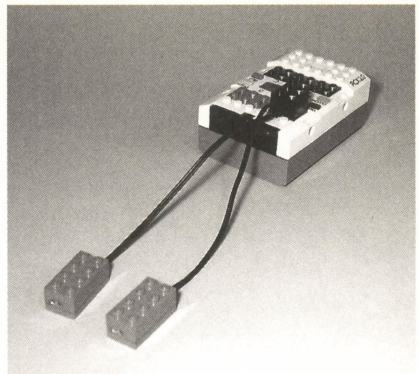
いときが60だとすると，2個つなげると，暗いときが50，明るいときが90ぐらいになります。

しかし，ライトセンサはとても個体差が大きいので，このとおりの値になるとは限りません。といっても，値の差が激しいので，多少周りの明るさが変わるような場所でも，ライトセンサが使えるようになります。

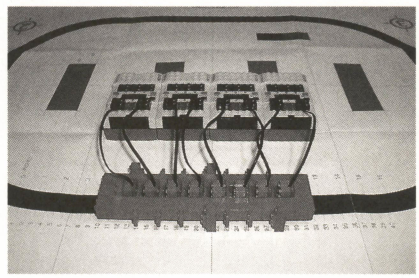
ところが，2個以上つけると変化が大きすぎて，値がすぐに100を超してしまい，使いづらくなってしまいます（**写真6**）。

## 個体差について

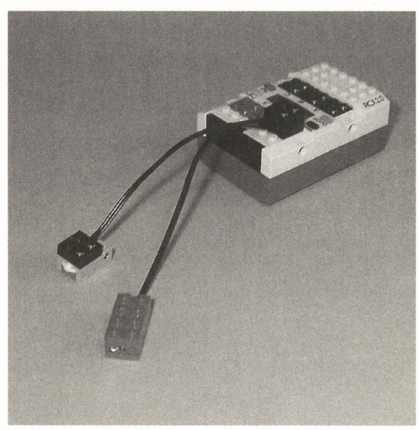
精密で有名なレゴ社ですが，ライトセンサの個体差，ローテーションセンサ（回転センサ）の誤差，RCXの個体差，RCXの液晶画面のゴミなどが多い物もあ



**《写真6》ライトセンサを2個つないだ状態。ちゃんとライトは光ります**



**《写真7》ライトセンサ8個をブロックに止めて，重石をつけて，黒い線に押さえつけています**



**《写真8》複合技。ちょっと意外な感じもあるけど，問題なし**

ります。とりあえず今回はライトセンサの個体差だけ紹介します。

今，手持ちのすべてのライトセンサの値を同じ条件（**写真7**）で測ると，**第2表**のようになります。かなりちがいますよね？ しかし，黒から白の変化は，ほとんどが19になっていて同じです。

## 複合技

じつはライトセンサとタッチセンサは複合技が可能です。たとえば**写真8**のように，タッチセンサとライトセンサのコードを接続して，ライトセンサモードで値を見ます。すると，タッチセンサが押

**《第2表》ライトセンサの黒，緑，白のときの値**

| センサ | 黒 | 緑 | 白 |
|---|---|---|---|
| 1 | 41 | 45 | 61 |
| 2 | 39 | 43 | 58 |
| 3 | 36 | 41 | 55 |
| 4 | 36 | 42 | 54 |
| 5 | 34 | 38 | 53 |
| 6 | 32 | 36 | 49 |
| 7 | 29 | 33 | 46 |
| 8 | 25 | 30 | 44 |

されない状態で，ライトセンサの値が表示されます。しかし，タッチセンサが押されているときは，値が「100」のまま変わりません。普通は，ライトセンサだけ接続した状態では，絶対といっていいほど100という値にはなりません。

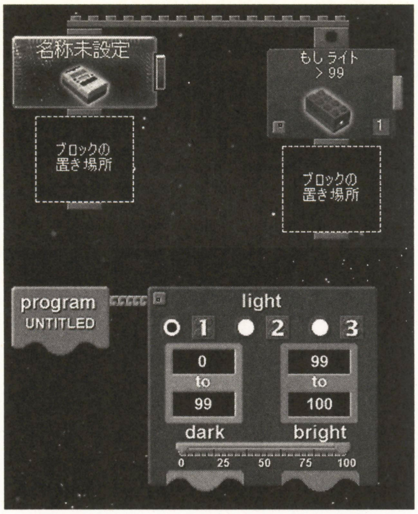
そこで，これを活用するのです。0～99までをライトセンサの値としてみて，タッチセンサを100未満と100で，「押されていない」「押された」というふうにわけると，センサポートの節約ができます（**第1図**）。これは，作例で実際に作ってみるので挑戦してみてください。

## ローテーションセンサ

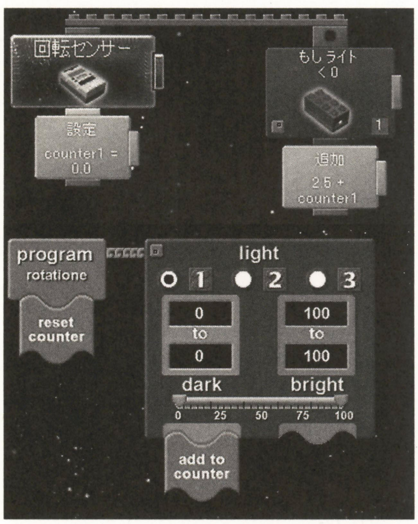
このローテーションセンサ（回転センサ）は，1回転につき16カウントの値が出されます。つまり，360÷16で22.5度の精度となります。しかし，現実ではなかなかうまくいきません。なぜうまくいかないかというと，RCXのファームウエアがよくないのと，センサ自体に無理があるためです。

一般的に工業製品やロボット相撲などで使われる距離センサ（回転数で距離を測るセンサ）は，「01010101」という値を繰り返し出しています。それを2個微妙にずらして配置してあり，それを1組として，回転数を読み取っているのです。

しかし，MINDSTORMSのローテーションセンサは，「01230123」という値を出しています。一見なにも問題がないのですが，ファームウエアが遅いためか，回転が速いと，値を飛び越して読んでしまいます。すると，RCXが勘違いして，とてつもない値を表示する場合が多くあります。またその逆でゆっくり回すと，各値の境目を行き来してしまい，これまた値が狂います。




《第1図》タッチセンサとライトセンサをあわせて使うときのプログラム

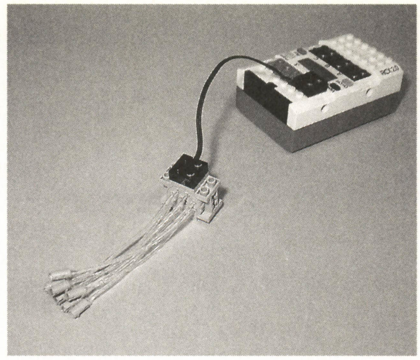



《第2図》ローテーションセンサの高精度版。1回転で4カウントする。値が0になったときにカウンタを1個上げる方法をとっている

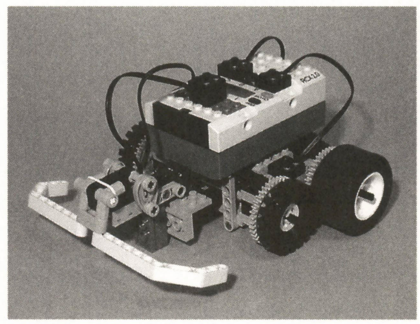
そこで裏技を使います。角度はわかりませんが，回転数が正確にわかるようになります。じつはローテーションセンサも，ライトセンサモードで値を見ることができるのです。

それを利用して**第2図**のようなプログラムを作ると，1回転ごとに4カウントされていきます。こうすることによって値の飛びや戻りをなくします。しかし，回転方向や，何回転進んで何回転戻ったかというのもわからなくなってしまいます(プログラムで工夫すればできるかもしれませんが…)。

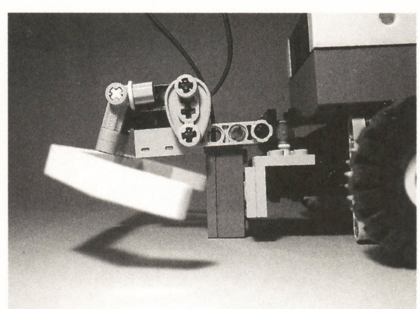
また，オプティカルファイバーを利用しても角度や回転数を測ることができます（**写真9**）。この場合は，ライトが光るときだけライトセンサモードで約80の値を出します。



《写真9》オプティカルファイバをつないだ状態。LEDが光っているので，ライトセンサモードでも電流は十分



《写真10》改造型ローバーボット。シングルバンパーの下にライトセンサがついている

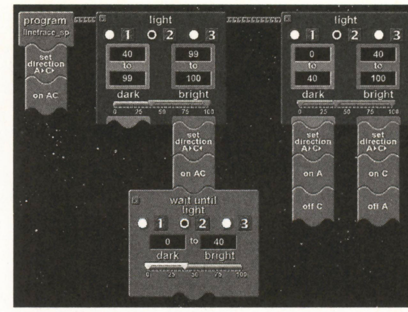


《写真11》ライトセンサ部分のアップ。90度曲がったプレート2枚と2×6のプレート1枚でライトセンサをつけている

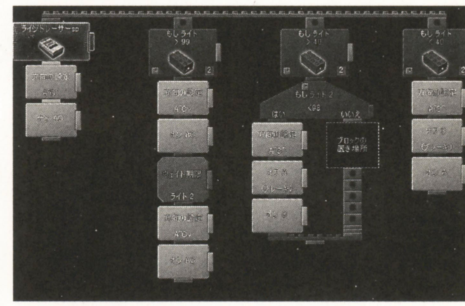
## 作例

今回は，ちょっと変わったライントレーサを作ってみたいと思います。どんなものかというと，「ぶつかると逆方向にトレースし始めるライントレーサ」です。普通なら，センサポートを2個使ってタッチセンサとライトセンサを使うのですが，今回紹介したセンサの並列つなぎでプログラムしてみましょう。

マシン自体は，カンタンなもので，ライントレースをするマシンにタッチセンサのバンパをつけるだけです。今回は組み立て図を省略するので，RIS1.5，




《第3図》RIS1.5のプログラム



《第4図》RIS2.0のプログラム

RIS2.0の説明書の最初に載っているローバーボットを使います。ローバーボットの前にライトセンサをつけるだけでいいのです（**写真10，11**）。

作例ではタイヤを使っていますが，キャタピラを使っても，足を使っても動きます。みなさんもどんどん工夫してみてくださいね。

さて，肝心のプログラムなのですが，**第3図，第4図**のような感じになります。ライトセンサモードを，0〜40，40〜99，100の3種類を監視していて，0〜40なら黒い線の上，40〜99なら白い部分の上，100ならタッチセンサが押された状態というふうにしています。

ここで，「40」という値が黒い線と白い部分の境の値としていますが，これはセンサによって変えないと動かないことも多いので，自分でセンサの値を変えて試してください。

動作の内容は，「ライントレースをする」「壁がないか確かめる」「壁があったらライントレースを止めて方向転換する」の三つです。

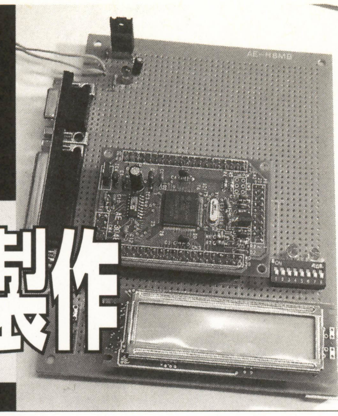
## 次回は

さてさて，次回はタイヤについてのお話です。たかがタイヤ，されどタイヤ。とても奥深くおもしろいものです。車系のマシンには切っても切れないものです。

松原拓也の自作一代記　連載9回目

# マイコンでBASICが動く!

# 「AKI-H8マイコンボード」の製作





AKI-H8/3048Fマイコンボード
7,800円
(アセンブラ，BASICコンパイラ付属)

秋月電子通商
http://www.akizuki.ne.jp/

今回は秋月電子通商の「AKI-H8マイコンボード」という電子工作キットを組み立てます(「マイコンボード」というのはイロイロな用途に使える組み込み用のコンピュータを載せたボードのことです)。プロセッサには日立製の「H8/3048F」という16bitCPUが使われています(**写真1，2**)。性能は8bitのワンチップマイコン(AVRやPIC)をはるかに上回り，動作クロックは16MHz(昔のパソコン並み)，フラッシュROMの容量はなんと128Kバイトです。このため，「H8」シリーズは複雑・大規模な計算を要するロボットの製作などでよく利用されています。

このキットには「マザーボード」という親基板が付属します。マザーボードにはスイッチ(タクトスイッチ4個，8Pディップスイッチ1個)と液晶パネルとROMライタが搭載されていて，キットを買ってすぐに使うことができます。マイコンボードとマザーボードは着脱可能になっていますので，取り外して別の機器に使いまわすこともできます。

なお，このキットには電源と通信ケーブルが付属しませんので，別途用意しておきましょう。

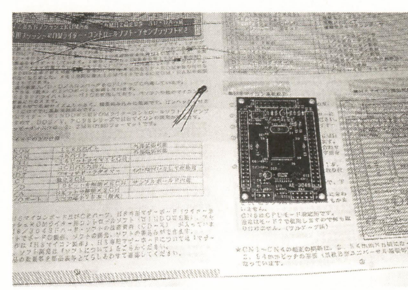
## マイコンボードの組み立て

まず，マイコンボードから作ります。部品配置図と基板上の印刷に書かれている番号(例：R1)と，部品表の型番(例：抵抗4.7kΩ)を照らし合わせてはんだ付けします(**写真3**)。

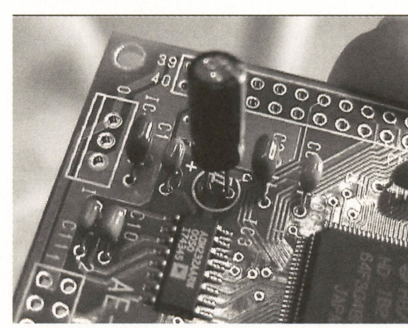
部品は背の低いものから，はんだこてを使ってはんだ付けしていきます。電解コンデンサやダイオードには極性がありますので向きに注意しましょう(**写真4**)。ピンヘッダはカッターナイフで切断します(**写真5**)。最後にピンヘッダをつければマイコンボードの完成です(**写真6**)。

## マザーボードの組み立て

つぎにマザーボードを組み立てます。組み立て方法はマイコンボードと同じで，部品表を見ながら取りつけていくだけです(**写真7**)。はんだ付けが完成したらマ



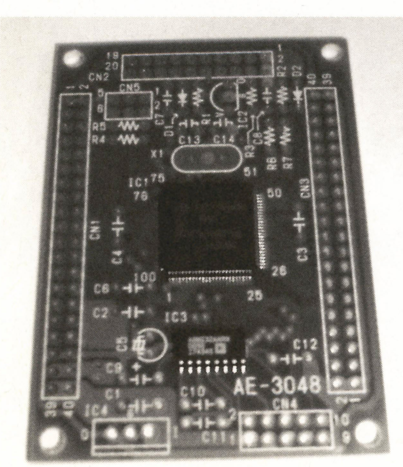
▲《写真3》基板と部品表を見比べながら，部品をはんだ付け



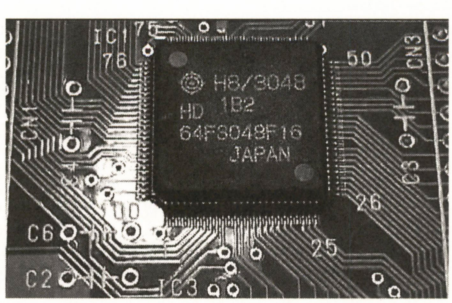
▲《写真4》部品の取りつけ時には極性をまちがえないように注意



▲《写真5》カッターナイフでピンヘッダを切断




◀《写真1》マイコンボード用の基板。CPUは実装済み




▲《写真2》日立製の16bitCPU「H8/3048F」。ROM128Kバイト，RAM4Kバイト，動作クロック16MHz，I/Oポート数最大78個

イコンボードのピンヘッダをマザーボードのピンフレームの上に載せます（**写真8**）。

あと，液晶パネル（**写真9**）も同様にマザーボードと合体させます。

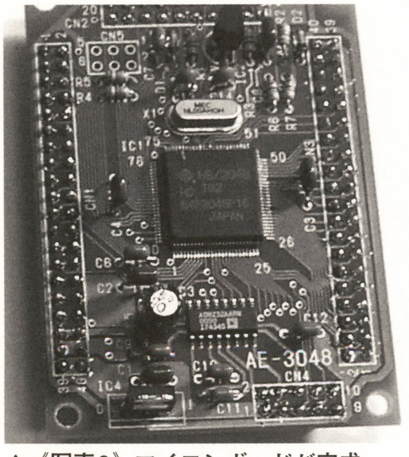
## 電源の入手

電源は14V以上，200mA以上のものが必要です。そこで，手近にあったACアダプタ（15V，最大800mA）を流用しました。それと，ACアダプタを接続するために2.1mmのジャックを購入しました（**写真10**）。これでハードウエアは完成です（**写真11**）。

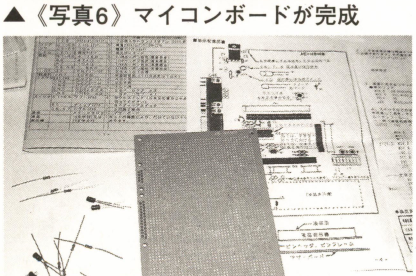
あとはスイッチをONにして電源を入れるだけですが，最初の状態ではフラッシュROMが空っぽなので，何も動きません。そこで，“MBTEST”というテストプログラムをROMに書き込んでみましょう。“MBTEST”は付属のCD-ROMに収録されています。

## パソコンとの接続

プログラムをフラッシュROMに書き込みます。まず，下準備として必要なファイルをCD-ROMからハードディスクにコピーします。



▲《写真6》マイコンボードが完成



▲《写真7》マザーボード側の組み立て

それから通信ケーブルでパソコンとマザーボードを接続します（**写真12**）。

## アセンブル方法

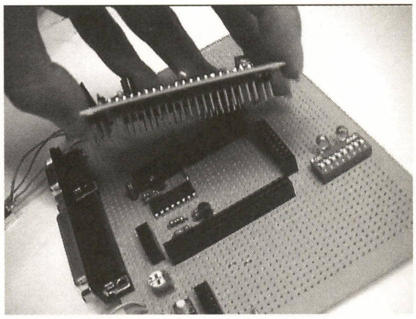
ソース・ファイル（拡張子はMAR）をアセンブルして，バイナリ・ファイル（拡張子はMOT）を作成しなくてはいけません。DOSプロンプトを立ち上げて，以下のコマンドを順に入力します（**写真13**）。対応機種はWindows98です。

**>A38H MBTEST**

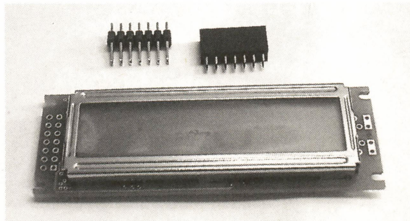
アセンブラ“A38H.EXE”でソース・ファイル（MAR）をオブジェクト・ファイル（OBJ）にします。

**>L38H MBTEST**

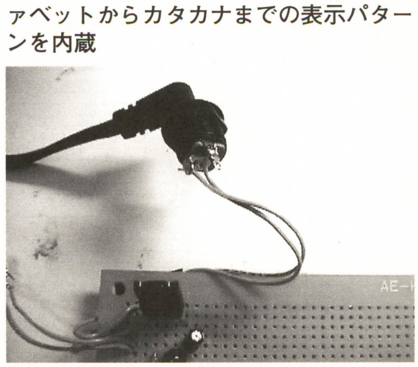
リンカ“L38H.EXE”を使ってロードモジュール・ファイル（ABS）にします。



▲《写真8》マイコンボードとマザーボードを合体



▲《写真9》16文字×2行の液晶（LCD）モジュール「SC1602BSLB/B」。アルファベットからカタカナまでの表示パターンを内蔵



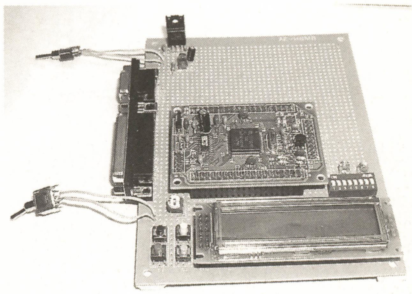
▲《写真10》ACアダプタ用のジャック「2.1ジャックMJ10」（東急ハンズで90円）

**>C38H MBTEST**

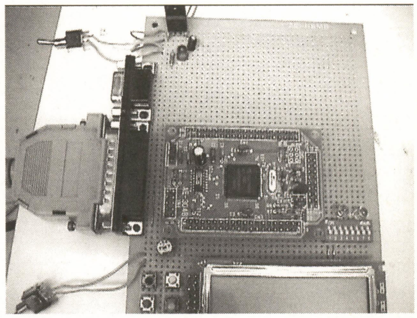
オブジェクトコンバータ“C38H.EXE”を使って，バイナリ・ファイル（MOT）を生成します（**写真14**）。

最後にROMライターソフト“FLASH.EXE”を使ってフラッシュROMに書き込みます（**写真15，16**）。フラッシュROMは最低100回まで書き換えることが可能です。書き換え回数が少なめですが，かわりに「プログラムをRAMにダウンロードしてデバッグできる」という便利な機能があります。

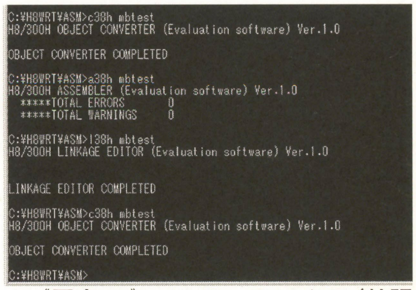
書き込み終わったら電源を切り，通常モードにしてから，再度電源を入れれば



▲《写真11》完成した「AKI-H8マイコンボード」



▲《写真12》コネクタ（Dsub25ピン）に通信ケーブルを接続。ライターモードにして電源を入れれば準備OK



```
C:¥H8WRT¥ASM>c38h mbtest
H8/300H OBJECT CONVERTER (Evaluation software) Ver.1.0

OBJECT CONVERTER COMPLETED

C:¥H8WRT¥ASM>a38h mbtest
H8/300H ASSEMBLER (Evaluation software) Ver.1.0
  *****TOTAL ERRORS        0
  *****TOTAL WARNINGS      0

C:¥H8WRT¥ASM>l38h mbtest
H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0

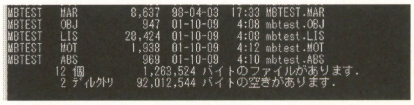

LINKAGE EDITOR COMPLETED

C:¥H8WRT¥ASM>c38h mbtest
H8/300H OBJECT CONVERTER (Evaluation software) Ver.1.0

OBJECT CONVERTER COMPLETED

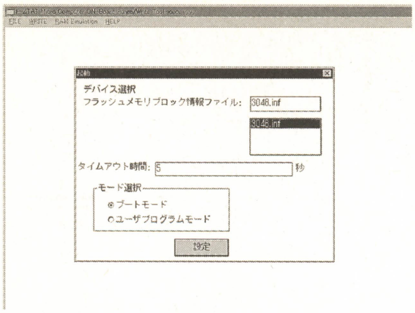
C:¥H8WRT¥ASM>
```

▲《写真13》ソース・ファイル（拡張子はMAR）をアセンブル

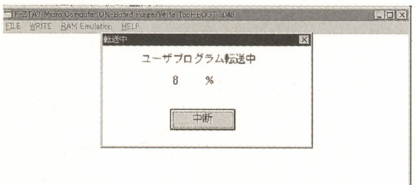


```
MBTEST   MAR        8,637  98-04-03  17:33 MBTEST.MAR
MBTEST   OBJ          947  01-10-09   4:08 mbtest.OBJ
MBTEST   LIS       28,424  01-10-09   4:08 mbtest.LIS
MBTEST   MOT        1,938  01-10-09   4:12 mbtest.MOT
MBTEST   ABS          969  01-10-09   4:10 mbtest.ABS
        12 個          1,263,524 バイトのファイルがあります.
         2 ディレクトリ   92,012,544 バイトの空きがあります.
```

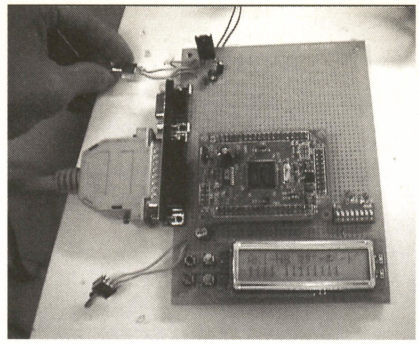
▲《写真14》最終的にバイナリ・ファイル（拡張子はMOT）が生成される



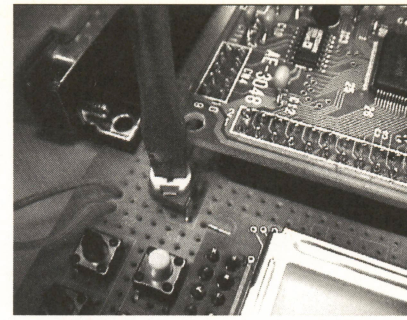

▲《写真15》ROMライタープログラムの起動画面




▲《写真16》フラッシュROMに書き込み中



▲《写真17》通常モードで電源を入れてプログラム実行。見事，動作成功!!



▲《写真18》ボリュームを回転させて液晶の濃さを調整

動き出します（**写真17**）。液晶に「AKI-H8」と表示されれば成功です。もし液晶に何も出ないときは液晶の濃さをボリュームで調節してみましょう（**写真18**）。

## BASICでプログラムできる!

このキットにはナンと「BASICコンパイラ」（**写真19**）が付いてきます。書式は「N-88BASIC」に似ていて，親しみやすいものになっています（**第1表**）。「C言語やアセンブラは難しい」という人にオススメです（**写真20，21**）。

## ゲームを作ってみる

手始めにカシオの「ゲーム電卓」風のゲームを作ってみました（**リスト1**）。

画面の右端からやってくる数字が「敵」です。その反対の左端が「自機」です（**写真22**）。

スイッチSW1/2を押すと自機の値（1

**《第1表》BASICコンパイラの予約語一覧表**

| | | | | | | |
|---|---|---|---|---|---|---|
| ABS | DATA | INPUT$ | NOT | RANDOMIZE | SPACE$ | TIMERSET |
| ADINP | DIM | INSTR | ON | READ | START | VAL |
| AND | ELSE | IRQ | OR | REM | STEP | VARPTR |
| ASC | END | LEFT$ | PEEK | RESTORE | STOP | WAIT |
| CALL | FOR | LEN | PEEKW | RETURN | STR$ | WHILE |
| CHR$ | GOSUB | LET | POKE | RETURNIRQ | STRING$ | WEND |
| CONSOLE | GOTO | MID$ | POKEW | RIGHT$ | THEN | XOR |
| CONTINUE | HEX$ | MOD | PORTSET | RND | TIMER | |
| DAOUT | IF | NEXT | PRINT | SERIALSET | TO | |

**《リスト1》プログラム・リスト“game.bas”**

```
'
'
        POKE &HFFFFC1, &H00    'ポート2を入力に設定
        POKE &HFFFFD8, &HFF    'ポート2をプルアップに設定

        POKE &HFFFFC5, &H00    'ポート4を入力に設定
        POKE &HFFFFDA, &HFF    'ポート4をプルアップに設定

        POKE &HFFFFC8, &HFF    'ポート5を出力に設定

        CONSOLE 0              '2行モードに

        TEKI$="______________" '14個
        MYCODE = &H31
        TEKICNT = 0 '敵出現カウンタ
        GANMA = 0 '難易度上昇カウンタ
        NANDO = 40 'ゲーム難易度初期値
        SCORE = 0 'スコア
:!LOOP
        G$ = LEFT$(TEKI$,1)
        IF G$ <> "_" THEN DISP$ = "   GAME OVER!   ":GOTO !LOOP50

        SW = PEEK(&HFFFFC7)    'ボタンの状態を取得
        S1 = SW AND &h10    'ボタン1...押されていれば0
        IF S1 <> 0 THEN GOTO !LOOP10
              MYCODE = MYCODE - 1
              IF MYCODE < &H31 THEN MYCODE = &H39
:!LOOP10
        S2 = SW AND &h20    'ボタン2...押されていれば0
        IF S2 <> 0 THEN GOTO !LOOP20
              MYCODE = MYCODE + 1
              IF MYCODE > &H39 THEN MYCODE = &H31
:!LOOP20
        S3 = SW AND &h40    'ボタン3...押されていれば0
        IF S3 <> 0 THEN GOTO !LOOP30
              I = 1
           NEWTEKI$=""
:!LOOP25
              TEKICODE$ = MID$(TEKI$,I,1)
              IF TEKICODE$ <> "_" THEN GOTO !LOOP26
                    NEWTEKI$ = NEWTEKI$ + TEKICODE$
                 I=I+1
                    IF I>14 THEN GOTO !LOOP30 ELSE GOTO !LOOP25
:!LOOP26
              ADDCODE = ASC(TEKICODE$) + MYCODE
              IF ADDCODE <> &H6A THEN GOTO !LOOP30
                    TEKICODE$ = "_"
:!LOOP27
                    NEWTEKI$ = NEWTEKI$ + TEKICODE$
                    I=I+1
                    IF I>14 THEN GOTO !LOOP29

                    TEKICODE$ = MID$(TEKI$,I,1)
                    GOTO !LOOP27
:!LOOP29
              TEKI$ = NEWTEKI$
:!LOOP30
        GANMA = GANMA + 1
        IF GANMA < 30 THEN GOTO !LOOP35
              GANMA = 0
              NANDO = NANDO - 1
              SCORE = SCORE + 1
:!LOOP35
        TEKICNT = TEKICNT + 1 '敵出現カウンタ+1
        IF TEKICNT < NANDO THEN GOTO !LOOP40
              TEKICNT = 0
              TEKI$ = RIGHT$(TEKI$,13) 'スクロール
              CH = RND(9) + &h31 '1~9までの乱数
              TEKI$ = TEKI$ + CHR$(CH)
:!LOOP40
        DISP$ = CHR$(MYCODE)
        DISP$ = DISP$ + ">"
        DISP$ = DISP$ + TEKI$
:!LOOP50
        DISP$ = DISP$ + "SCORE:"
        DISP$ = DISP$ + STR$(SCORE)
        PRINT DISP$

        WAIT 100

        GOTO !LOOP
```

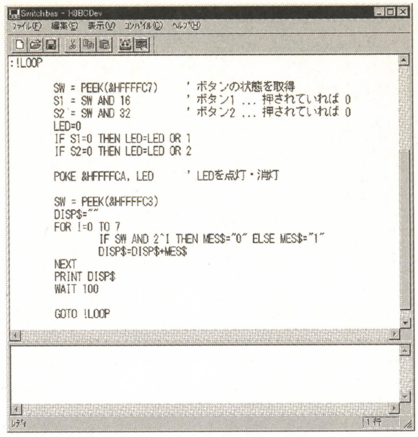



▲《写真19》付属する「BASICコンパイラ」のCD-ROM

〜9）がループします。スイッチSW3を押したとき，左側の数と右側の数が足して「10」になると敵が消滅します。敵が左端に来るとゲーム・オーバーです（写真23）。

## プログラムの解説

液晶への表示は「PRINT」命令，I/Oの読み込みは「PEEK」命令です（N88-BASICでおなじみ）。分岐は「IF THEN」




▲《写真20》BASICでプログラミング中

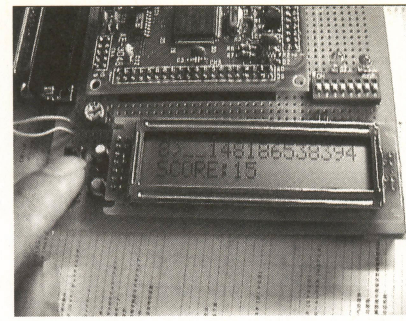



▲《写真21》BASICプログラムをコンパイル。成功するとMOTファイルが生成される

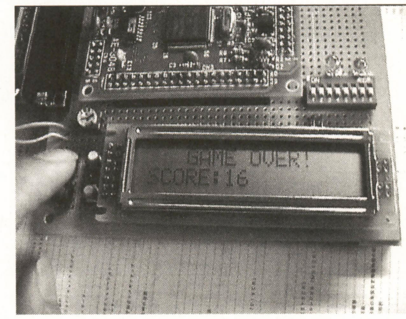
なのですが「ENDIF」がないのでしかたなく「GOTO」を多用しています。そのため，ちょっとソースが見づらいです。

## 最後に

このプログラムはわずか2Kバイト。




▲《写真22》右端からやってくる敵（数字）を消していくゲームです

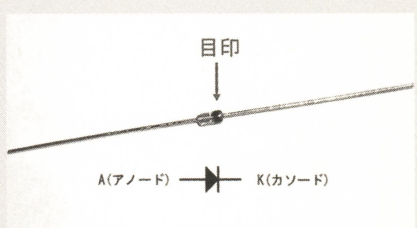



▲《写真23》敵が左端に来てしまうとゲーム・オーバー

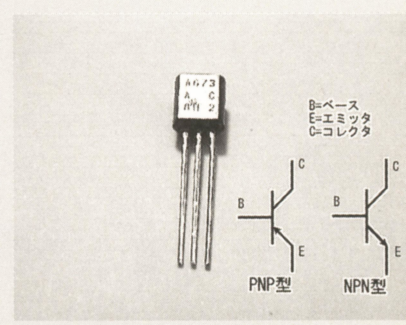
ROMの全容量の1.5%しか使ってません。H8の潜在能力を完全に生かせれば，もっともっと凄いプログラムができると思います。

## 電気工作の基礎知識

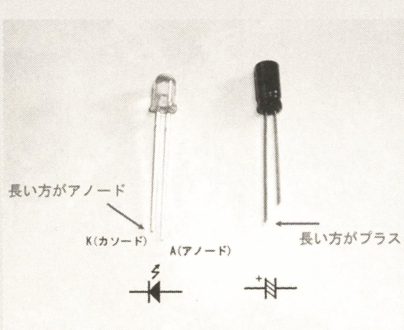
**いままでの復習になりますが，部品の機能と見分けかたを紹介します。**



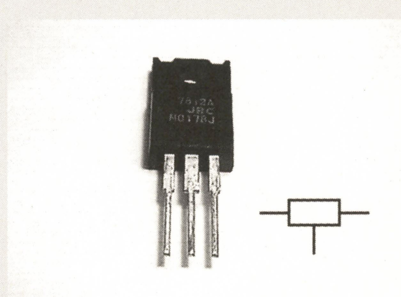

▲ダイオードは目印のあるほうがカソード側です。電流はアノードからカソードへ一方通行に流れます



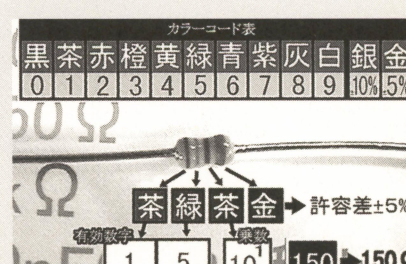

▲トランジスタは電流の流れを増幅する部品です。ピン配置は左からE，C，B（2SA1015の場合）




▲発光ダイオード（左）は電流を流すと発光する部品です。足の長いほうがアノード。電解コンデンサ（右）は電気をためる部品です。足の長いほうがプラス側になります



▲三端子レギュレータは電圧を調整する部品です。ピン配置は左からIN，GND，OUT（LM7812の場合）




▲抵抗は電流に負荷を与えるものです。抵抗値は色のついた線から読み取ります

デジタルとアナログの境界線上で作る電子オモチャ

テクノ・トイズ・ラボラトリ
Techno Toys Laboratory

今月の製作

# ポケットライト・デラックス 後編

## テクノ・オモチャ研究所

TEXT&製作●クワクボリョウタ



デバイス・アーチスト。1998年頃から「ビット」をキーワードにメディアアート作品や電子オモチャなどを製作・発表。新宿のICCでの展覧会「信用ゲーム」に出品してます（11/2〜12/24）。ヨロシクどうぞ。
www.vector-scan.com

**今月はポケットライト・デラックスの製作の後編だ。前回はフルカラーLEDの多色表示制御の部分を実験したけど，ポケットライトとして実用するにはパワーのON/OFF制御が必要だ。回路にはすでにスイッチが一つ付けてある。今回はそれを使ってパワー管理だけでなく，発光色の選択を行えるようにプログラムしてみよう。**

### 完成形の構想

当研究所のポリシーは最終的にオモチャとして見せびらかせられる状態に完成すること！　単にフルカラーLEDがキレイにグラデーション動作するだけでは飽き足らないのだ。そこで今回はポケットライトとして使えるようにケースの加工をして，残るスイッチの動作をプログラムしていこうと思う。

### 工作の時間です

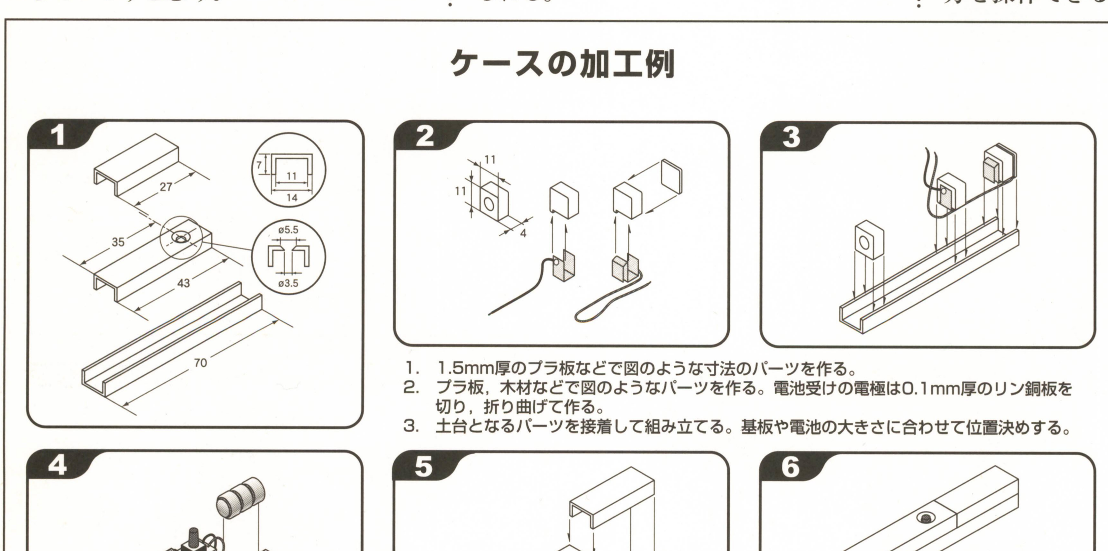
最初にケースの加工について説明しておこう。加工例は**第1図**のとおりだ。これを参考に各自入手しやすい素材で作ってみよう。この寸法で作ると，タクト・スイッチのボタン部分がちょうどケースの面とそろって出っ張らないので，持ち運び時に誤って押してしまうことが避けられる。

チェーンをつけてキーホルダーなんかにするのも良いかもしれないね。完成品の外観は**写真1**，内部のようすは**写真2**のようになっている。

### プログラムの追加

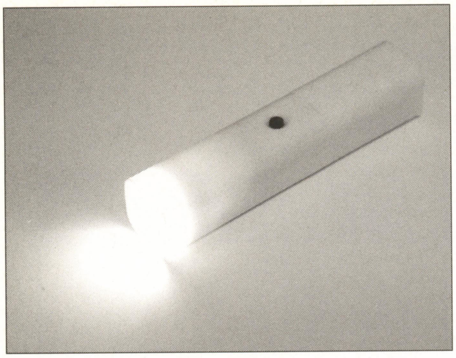
前回製作した回路基板にはタクト・スイッチが一つのっている。これを使ってパワーのON/OFFと発光色の選択の両方を操作できるようにしよう。一つのスイッチに二つの機能を持たせることってできるんだろうか？　スイッチ一つでは1ビットの情報しか扱えないからそれは無理…だと思う？　いやいや，そんなことはないよ。単純なスイッチのON/OFF操作に「押している時間の長さ」という要素を加えることで，たった一つのスイッチでもいろんなことができてしまうんだ。

**第2図**がポケットライト・デラックスのスイッチ操作の大雑把なイメージだ。パワーON時に長押し（1秒く




1. 1.5mm厚のプラ板などで図のような寸法のパーツを作る。
2. プラ板，木材などで図のようなパーツを作る。電池受けの電極は0.1mm厚のリン銅板を切り，折り曲げて作る。
3. 土台となるパーツを接着して組み立てる。基板や電池の大きさに合わせて位置決めする。
4. 電源のコードを基板にハンダ付けし，ケースに収納する。
5. 接着剤で基板のフタを取り付ける。
電池のフタはテープなどで着脱可能にする。
6. 完成！

《第1図》これを参考に，自分なりのケースを作ってみよう



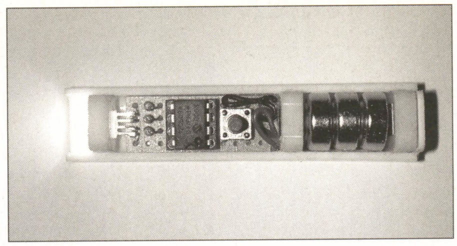
《写真1》完成したポケットライト・デラックスをフルパワーで点灯したときのようす

らい押しっぱなし）すると，前回のプログラムのように発光色が変化し始め，もう一度クリックすることで色をロックさせる仕様になっている。こうすることでスイッチ一つでの色の選択を可能にしている。

## 状態遷移図で考える

以上の処理をするルーチンをここではコントロール・ルーチンと呼ぶことにする。このルーチンの動作をプログラムに落とし込むために詳しく書き直したのが**第3図**だ。

三つの丸がシステムの状態（ステータスともいう）を表す。OFF（パワーOFF），ON（パワーON，発光色固定），CYCLE（発光色変化）の三つの状態がある。システムは常にこの三つのうちのどれかの状態にある。そして特定の条件を満たすと矢印に沿って他の状態に遷移（なんだか難しい言葉だけど，移動と同じような意味の用語）する。
遷移の引き金となる条件は，当然スイッチの操作によって発生される。矢印の根元に書いてあるのが，その条件だ。この



《写真2》これが内部のようすだ。基板と電池をコンパクトに収納した

条件（出来事）のことをイベントという。

注目してほしいのは，ONからは条件によってOFFまたはCYCLEのどちらかに分岐する点だ。長押しすればCYCLEになるし，クリックすればOFFになる。また，この図によれば，OFFから直接CYCLEになったり，逆にCYCLEから直接OFFになったりはしないことが分かる。

このような図は状態遷移図と呼ばれる。このようにして考えると制御の全体像をつかんだり，予期せぬ状況を発見したりできて便利だ。

## 各イベントの定義

さて，「長押し」や「クリック」というイベントについて厳密に定義してみよう。それはこのシステムで「クリック」と「長押し」をどのように判別するかを考えることでもある。

マイコンの立場になって想像してみてほしいのだけど，スイッチが押された瞬間には，これからユーザが長押ししたいのか，それともクリックするつもりなのかはまだ分からない。スイッチが押され続けて一定時間が経過したときにはじめて「長押し」されたと判断される。だからスイッチが押された瞬間にタイマをリセットして経過時間をはかる仕組みが必要になる。

一方，「クリック」はスイッチが押され，長押しと判定される以前にスイッチが開放されたときに判定される。だから，クリックが発生するのは押したときではなく，放したときだ（**第4図**）。

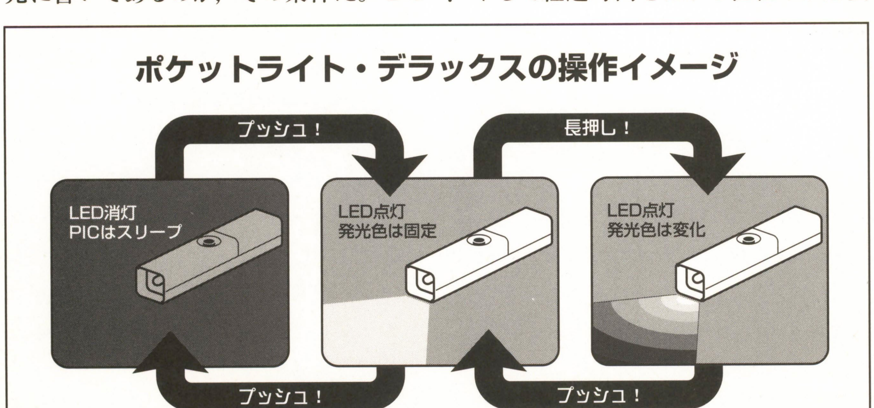
ところで，OFFからONに移行するイベントは「クリック」ではなく「押し下げ」と書かれている。つまり，パワーONはスイッチを押した瞬間に起きますよ，ということだ。これには理由がある。パワーOFFはPICのスリープ機能を使い，スリープからのウェイクアップ（つまりパワーON）は入力ピンの電位の変化を条件に設定する関係上，ウェイクアップが起きるのは，スイッチを押した瞬間とならざるをえないからだ。

スイッチの動作などをプログラムする際にはこのように細々したところまで考慮しないと思わぬ動作をしてしまうことがある。でも面倒くさいようでいて，人とマシンの接点だけになかなか面白い領域なんじゃないかな。

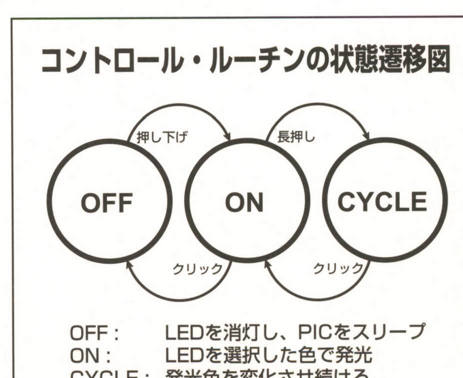
## イベントを発生する

以上のイベントはどうやって発生するの？　ということを説明しよう。「クリック」とか「長押し」という，ちょっと複雑なスイッチのふるまいは，より低レベルなイベント（スイッチの単純なON/OFFと，押された時間を測るタイマ）によって作られる。

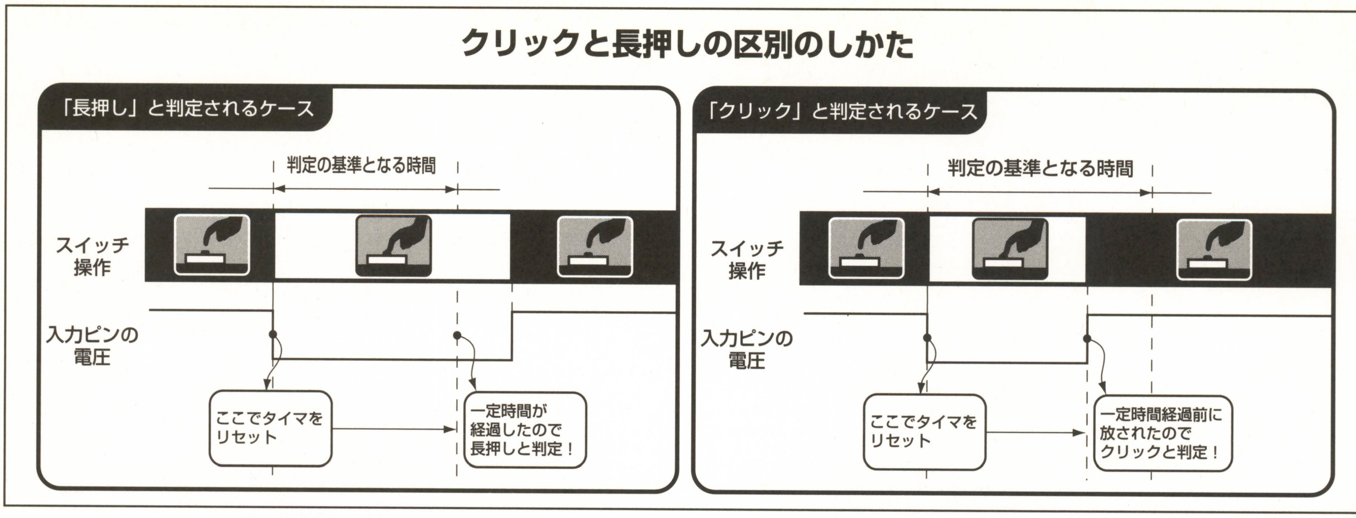
まずスイッチまわりの回路を確認しておこう（**第5図**）。タクト・スイッチはPIC12C509のGP3に接続されており（このピンは入力専用のピン），内部でプルアッ



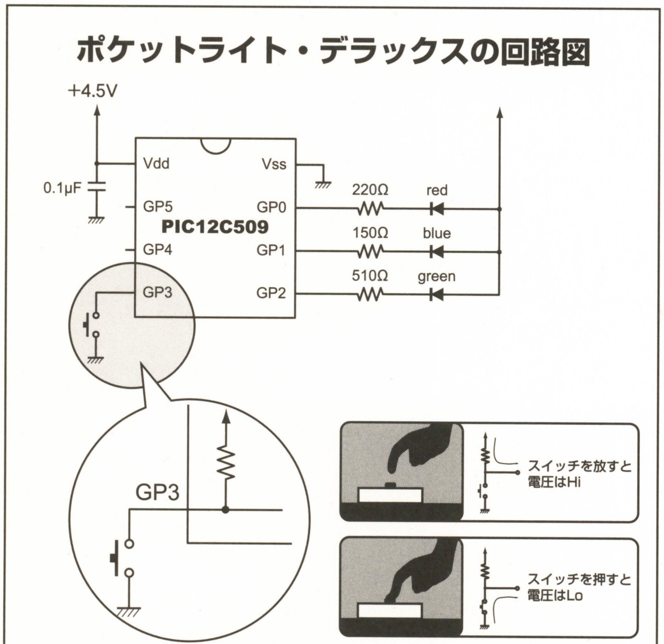

《第2図》操作は簡単にプッシュと長押しでできる



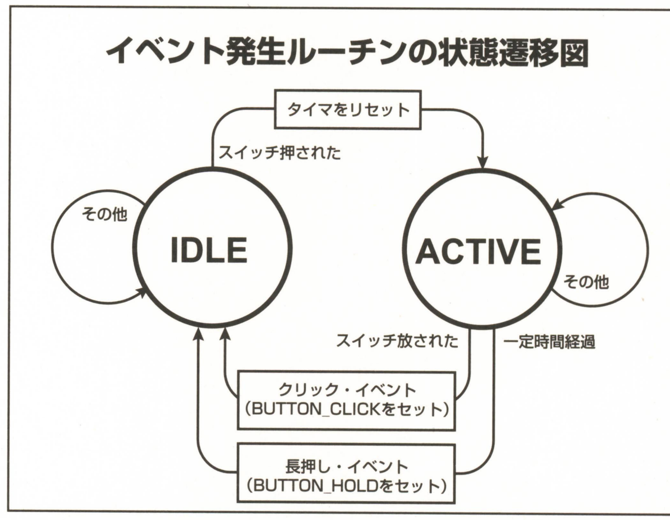

《第3図》第2図を詳しく書き直したもの。三つの丸が状態。矢印の根もとに書いてあるのが，イベントだ




《第4図》基準となる時間より長く押され続けたら“長押し”，それ以前に放されたら“クリック”と判断する




《第5図》GP3は内部でプルアップするようにプログラムで指定する（ソースリストの初期設定を参照のこと）。




《第6図》スイッチの入力の変化とタイマをもとに，クリックと長押しのイベントを発生させる仕組み

プするようにプログラムで設定している。そして，スイッチのもう一方の端子はグラウンドに接続されている。スイッチが開いているとピンの電圧は4.5V，閉じると0Vとなり，PICからはそれぞれ"1"と"0"として読み出される。押しているときに"0"の負論理となっていることに注意。

このスイッチが接続された入力ピンから読み出した値と，スイッチが押されている時間を測るタイマをもとに，クリック・イベント(BUTTON_CLICK)と，長押し・イベント(BUTTON_HOLD)を，各フラグを通じてコントロール・ルーチンに知らせる。これがイベント発生ルーチンの仕事だ。

スイッチの「押し下げ」についてはPICのウェイク・アップ機能が処理するので別扱いになっている。

**第6図**はイベント発生ルーチンの状態遷移図だ。このルーチンは実行ごとに入力ピンの値を読み出し，前回読んだ値と比較することにより，スイッチの変化を検出する。はじめはIDLE状態で，スイッチがひとたび押されると，ACTIVE状態に移る。ACTIVE状態では実行ごとにタイマを動作させ，一定時間ACTIVE状態にとどまった場合（タイムアウト）は長押しと判定し，BUTTON_HOLDイベントのフラグを立てる。一方，一定時間経過前にスイッチが開放されたらBUTTON_CLICKイベントのフラグを立てる。どちらの場合もイベント発生後はIDLE状態に戻る。

こうして発生したイベントは先に説明したコントロール・ルーチンが受け取り，パワーON/OFFや色選択の動作を引き起こす。つまりポケットライト・デラックスのコントロールは2段構えの状態遷移によって構成されているわけだ。

一見まわりくどいようだけど，一気に制御プログラムを書くより，このようにいくつかのレイヤに分けて考えたほうがずっと効率的だし，変更も容易だ。サブルーチンにしておけば，他のアプリケーションに使いまわしすることもできる。

## 全体の流れ

以上のルーチンと，前回作ったルーチンをつなげたソース・リストが**リスト1**だ。

LEDのPWM処理や輝度更新（前回プログラムした部分）とスイッチの処理は交互に行う。処理は約4msecで一周するので，スイッチの状態を監視する間隔もこの周期となる。この周期であれば，チ

ャタリング（スイッチのバタつき）の影響はほとんどないので，チャタリング対策はしていない。

はじめにイベント発生ルーチンでスイッチの状態を処理し，続くコントロール・ルーチンで本機のパワーおよび発光色変化のモードを決定，つぎのPWM処理ルーチンでLEDの発光，CYCLE状態（発光色変化）のときは最後に輝度更新ルーチンでLEDの発光色を変える。

パワーOFFになるときには，スイッチが放されていることを確認して，SLEEP命令を実行している。再びスイッチが押されればすぐにウェイクアップして，メインループに戻る。パワーONの処理はこれだけだ。

ポケットライト・デラックスを作ってみて一番難しかったのはLEDを暗く発光させるときの輝度の変化だった。消えかかる寸前の明るさの変化はなかなか自然に見せるのは難しい。

この問題は，プログラムが複雑になるが多少の改善は可能だ。我こそはと思う人は是非挑戦してみてほしい！

《リスト1》ポケットライト・デラックス後編のリスト（アセンブラ）

```
;==================================================================
;
;                 ﾎﾟｹｯﾄﾗｲﾄ・ﾃﾞﾗｯｸｽ
;
;            c o d e d  b y  ｸﾜｸﾎﾞﾘｮｳﾀ
;
;                 PIC12C508 or 509
;                    +--------+
;          +4.5V ----|Vdd  Vss|---- GND
;                x---|GP5  GP0|---> Red cathode
;                x---|GP4  GP1|---> Blue cathode
;               SW --->|GP3  GP2|---> Green cathode
;                    +--------+
;
;==================================================================
        LIST    C=120, p=12C509, r=dec, x=off, n=0
        include <p12c509.inc>

;--------- 各レジスタの割り当て ---------
port_buf        equu    0x07            ; LEDのｽｲｯﾁﾝｸﾞﾊﾟﾀｰﾝ
t               equ     0x08            ; 発光時間比較ｶｳﾝﾀ
t_red           equ     0x09            ; 発光時間
t_green         equ     0x0a            ;   //
t_blue          equ     0x0b            ;   //
m_red           equ     0x0c            ; 輝度更新用ｶｳﾝﾀ
m_green         equ     0x0d            ;     //
m_blue          equ     0x0e            ;     //
p_red           equ     0x0f            ; 輝度更新用ﾎﾟｲﾝﾀ
p_green         equ     0x10            ;     //
p_blue          equ     0x11            ;     //
stat            equ     0x12            ; ｽﾃｰﾀｽ ﾌﾗｸﾞ用ﾚｼﾞｽﾀ
event           equ     0x13            ; ｲﾍﾞﾝﾄ ﾌﾗｸﾞ用ﾚｼﾞｽﾀ
current_pin     equ     0x14            ; 入力保持用
prev_pin        equ     0x15            ; 入力変化検出用
button_count    equ     0x16            ; ｽｲｯﾁ持続時間計測用

;------------- 各初期値 ---------------
M_RED_INIT      equ     19              ;
M_GREEN_INIT    equ     18              ; 各色輝度の更新周期
M_BLUE_INIT     equ     17              ;

BUTTON_COUNT_INIT       equ     120     ; 長押し判定の基準時間

;--------- GPIOのビットマップ -----------
BUTTON_PIN              equ     3       ; ｽｲｯﾁが接続されたピン

;--------- eventのビットマップ ----------
BUTTON_CLICK            equ     0       ; ｸﾘｯｸ ｲﾍﾞﾝﾄ ﾌﾗｸﾞ
BUTTON_HOLD             equ     1       ; 長押し ｲﾍﾞﾝﾄ ﾌﾗｸﾞ

;--------- statのビットマップ -----------
BUTTON_STAT             equ     0       ; ｲﾍﾞﾝﾄ発生ﾙｰﾁﾝの状態
                                        ; ( 0:IDLE / 1:ACTIVE )
COLOR_CYCLE             equ     1       ; ｺﾝﾄﾛｰﾙ ﾙｰﾁﾝの状態
                                        ; ( 0:ON / 1:CYCLE )

;==================================================================
;                   初期設定、リセット後の分岐
;==================================================================
        org     0x000
initialize:
        movwf   OSCCAL                  ; 内部RCの校正

        movlw   b'00001000'             ; I/Oの入出力方向設定
        tris    GPIO                    ;

        movlw   b'00010111'             ; ﾋﾟﾝ入力の変化でｳｪｲｸｱｯﾌﾟ
        option                          ; & 内部ﾌﾟﾙｱｯﾌﾟ使用

        ;-------- 電源投入後か、ｳｪｲｸｱｯﾌﾟ後かを判定 -------------
        btfsc   STATUS,GPWUF            ; ｳｪｲｸｱｯﾌﾟ後はwake_upへ
        goto    wake_up                 ; 電源投入後は以下へ
        ;-------------------------------------------------------

        movlw   b'00000111'             ; 全LEDをOFF
        movwf   GPIO                    ;

        clrf    stat                    ; 各ﾚｼﾞｽﾀの初期値を設定
        movlw   255                     ;
        movwf   t_red                   ;
        movwf   t_green                 ;
        movwf   t_blue                  ;
        movlw   128                     ;
        movwf   p_red                   ;
        movwf   p_green                 ;
        movwf   p_blue                  ;
        movlw   M_RED_INIT              ;
        movwf   m_red                   ;
        movlw   M_GREEN_INIT            ;
        movwf   m_green                 ;
        movlw   M_BLUE_INIT             ;
        movwf   m_blue                  ;

;==================================================================
;                          ｽﾘｰﾌﾟ処理
;==================================================================
power_down:
        btfss   GPIO,BUTTON_PIN         ; ｽｲｯﾁが放されている
        goto    $-1                     ; のを確認

        movlw   b'00000111'             ; 全LEDをOFF
        movwf   GPIO                    ;

        sleep                           ; スリープ
        nop                             ;

        ; sleep命令実行後、ｽｲｯﾁが押されるとｳｪｲｸｱｯﾌﾟが起き、
        ; 0x000から実行を再開する。
        ; ｳｪｲｸｱｯﾌﾟ後はGPWUFがｾｯﾄされているので、
        ; それを調べて、wake_upから実行を再開する。

;==================================================================
;                        ｳｪｲｸｱｯﾌﾟ処理
;==================================================================
wake_up:
        movf    GPIO,W                  ; ｽｲｯﾁの変化検出用
        movwf   prev_pin                ;     に状態を読み込む

;==================================================================
;                         ﾒｲﾝ ﾙｰﾌﾟ
;==================================================================
main:
        ;==========================================================
        ;                   ｲﾍﾞﾝﾄ発生ﾙｰﾁﾝ
        ;==========================================================
button_proc:
        clrf    event                   ; ｲﾍﾞﾝﾄﾌﾗｸﾞをｸﾘｱ

        movf    GPIO,W                  ; ｽｲｯﾁの状態読み取り
        movwf   current_pin             ;

        btfsc   stat,BUTTON_STAT        ;
        goto    button_proc_10          ;

        ;-------- ｽｲｯﾁがIDLE状態の場合 ------------------------
        comf    current_pin,W           ; ｽｲｯﾁ押されたら以下へ
        andwf   prev_pin,W              ;
        andlw   1<<BUTTON_PIN           ;
        btfsc   STATUS,Z                ;
        goto    button_proc_90          ; それ以外は90へ

        bsf     stat,BUTTON_STAT        ; ACTIVE状態にする

        movlw   BUTTON_COUNT_INIT       ; ｽｲｯﾁ持続時間のﾀｲﾏ
        movwf   button_count            ; をﾘｾｯﾄ

        goto    button_proc_90          ; 90へ

        ;-------- ｽｲｯﾁがACTIVE状態の場合 ----------------------
button_proc_10:
        btfsc   current_pin,BUTTON_PIN  ; ｽｲｯﾁが押されている
                                        ;           なら以下へ
        goto    button_proc_20          ; それ以外は20へ

        decfsz  button_count,F          ; ﾀｲﾏを動かす
                                        ; 0になったら以下へ
        goto    button_proc_90          ; それ以外は90へ

        bsf     event,BUTTON_HOLD       ; 長押しｲﾍﾞﾝﾄを発生
        bcf     stat,BUTTON_STAT        ; IDLE状態にする
        goto    button_proc_90          ; 90へ

button_proc_20:
        comf    prev_pin,W              ; ｽｲｯﾁが
        andwf   current_pin,W           ;  放されたら以下へ
        andlw   1<<BUTTON_PIN           ;
        btfsc   STATUS,Z                ;
        goto    button_proc_90          ; それ以外は90へ

        bsf     event,BUTTON_CLICK      ; ｸﾘｯｸｲﾍﾞﾝﾄを発生
        bcf     stat,BUTTON_STAT        ; IDLE状態にする

button_proc_90:
        movf    current_pin,W           ; 比較用にﾃﾞｰﾀを保存
        movwf   prev_pin                ;

        ;==========================================================
        ;                   ｺﾝﾄﾛｰﾙ・ﾙｰﾁﾝ
        ;==========================================================
main_control:
        btfsc   stat,COLOR_CYCLE        ; ON状態なら以下へ
        goto    main_control_10         ; CYCLE状態の場合は10へ

        ;-------- ON状態の場合 ---------------------------------
        btfsc   event,BUTTON_CLICK      ; ｸﾘｯｸされたら以下へ
        goto    power_down              ; それ以外はpower_downへ

        btfsc   event,BUTTON_HOLD       ; 長押しされたら
        bsf     stat,COLOR_CYCLE        ; CYCLE状態にする

        goto    main_control_90         ; 90へ

        ;-------- CYCLE状態の場合 ------------------------------
main_control_10:
        btfsc   event,BUTTON_CLICK      ; クリックされたら
        bcf     stat,COLOR_CYCLE        ; ON状態にする

main_control_90:

        ;==========================================================
        ;                   PWM処理ルーチン
        ;==========================================================
        clrf    t                       ; tを初期化
main_10:
        ;----------------------------------------------------------
        ; tとt_red,t_blue,t_greenを比較し、
        ; それぞれがtより大きい場合に各LEDを点灯する
        ; port_buf には '00000GBR' の並びで結果が入り、
        ; 最後にまとめてGPIOに出力する
        ; (I/Oはｶｿｰﾄﾞに接続されているので0出力で点灯)
        ;----------------------------------------------------------
        clrf    port_buf                ; port_bufをｸﾘｱ

        movf    t_green,W               ; if t < t_green
        subwf   t,W                     ; then C = 0
        rlf     port_buf,F              ; port_buf
                                        ; =(port_buf<<1)+C

        movf    t_blue,W                ; if t < t_blue
        subwf   t,W                     ; then C = 0
        rlf     port_buf,F              ; port_buf
                                        ; =(port_buf<<1)+C

        movf    t_red,W                 ; if t < t_red
        subwf   t,W                     ; then C = 0
        rlf     port_buf,F              ; port_buf
                                        ; =(port_buf<<1)+C

        movf    port_buf,W              ; 結果を
        movwf   GPIO                    ;    I/Oに出力する

        incfsz  t,F                     ; tが100hになるまで
        goto    main_10                 ;         繰り返し

        ;==========================================================
        ;                   輝度更新ルーチン
        ;==========================================================
        btfss   stat,COLOR_CYCLE        ; CYCLE状態なら以下へ
        goto    main                    ; それ以外はmainへ

        ;-------- 赤の輝度更新 ---------------------------------
red:
        decfsz  m_red,F                 ; 赤の更新ﾀｲﾐﾝｸﾞ？
        goto    green                   ;    違うならgreenへ

        movlw   M_RED_INIT              ; 更新ｶｳﾝﾀをﾘｾｯﾄ
        movwf   m_red                   ;

        movf    p_red,W                 ; t_red <-
        call    get_t                   ;    get_t(p_red++)
        bcf     STATUS,PA0              ;
        movwf   t_red                   ;
        incf    p_red,F                 ;

        ;-------- 緑の輝度更新 ---------------------------------
green:
        decfsz  m_green,F               ; 緑の更新ﾀｲﾐﾝｸﾞ？
        goto    blue                    ;    違うならblueへ

        movlw   M_GREEN_INIT            ; 更新ｶｳﾝﾀをﾘｾｯﾄ
        movwf   m_green                 ;

        movf    p_green,W               ; t_green <-
        call    get_t                   ;    get_t(p_green++)
        bcf     STATUS,PA0              ;
        movwf   t_green                 ;
        incf    p_green,F               ;

        ;-------- 青の輝度更新 ---------------------------------
blue:
        decfsz  m_blue,F                ; 青の更新ﾀｲﾐﾝｸﾞ？
        goto    main_90                 ;    違うなら90へ

        movlw   M_BLUE_INIT             ; 更新ｶｳﾝﾀをﾘｾｯﾄ
        movwf   m_blue                  ;

        movf    p_blue,W                ; t_blue <-
        call    get_t                   ;    get_t(p_blue++)
        bcf     STATUS,PA0              ;
        movwf   t_blue                  ;
        incf    p_blue,F                ;

main_90:
        goto    main                    ; くり返し

;==================================================================
;                  ﾃｰﾌﾞﾙ : 発光時間の変化
;
;           Wに0～255をセットしてget_tをコールすると
;           Wの値に応じたアドレスの値を返す
;           ("dt n" は "retlw n" と同じ意味)
;==================================================================
get_t:
        bsf     STATUS,PA0              ;
        movwf   PCL                     ;

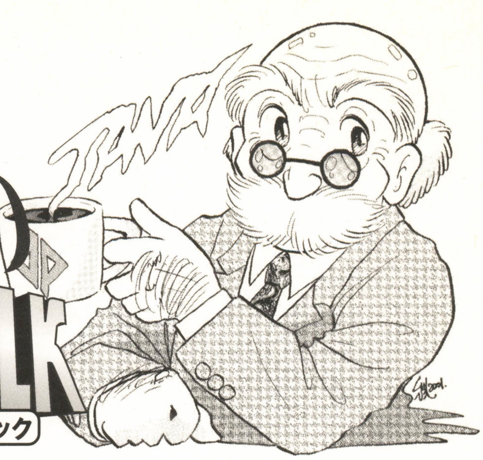
        org     0x200

        dt        1,   1,   1,   1,   1,   1,   1,   1
        dt        1,   2,   2,   2,   2,   2,   2,   2
        dt        2,   2,   2,   2,   2,   3,   3,   3
        dt        3,   3,   3,   3,   4,   4,   4,   4
        dt        4,   4,   5,   5,   5,   5,   5,   6
        dt        6,   6,   6,   7,   7,   7,   8,   8
        dt        8,   9,   9,   9,  10,  10,  11,  11
        dt       12,  12,  13,  13,  14,  15,  15,  16
        dt       17,  17,  18,  19,  20,  21,  22,  23
        dt       24,  25,  26,  27,  28,  29,  31,  32
        dt       33,  35,  36,  38,  40,  41,  43,  45
        dt       47,  49,  51,  54,  56,  59,  61,  64
        dt       67,  70,  73,  76,  79,  83,  86,  90
        dt       94,  98, 103, 107, 112, 117, 122, 128
        dt      133, 139, 145, 152, 158, 165, 173, 180
        dt      188, 197, 205, 214, 224, 234, 244, 255
        dt      255, 244, 234, 224, 214, 205, 197, 188
        dt      180, 173, 165, 158, 152, 145, 139, 133
        dt      128, 122, 117, 112, 107, 103,  98,  94
        dt       90,  86,  83,  79,  76,  73,  70,  67
        dt       64,  61,  59,  56,  54,  51,  49,  47
        dt       45,  43,  41,  40,  38,  36,  35,  33
        dt       32,  31,  29,  28,  27,  26,  25,  24
        dt       23,  22,  21,  20,  19,  18,  17,  17
        dt       16,  15,  15,  14,  13,  13,  12,  12
        dt       11,  11,  10,  10,   9,   9,   9,   8
        dt        8,   8,   7,   7,   7,   6,   6,   6
        dt        6,   5,   5,   5,   5,   5,   4,   4
        dt        4,   4,   4,   4,   3,   3,   3,   3
        dt        3,   3,   3,   2,   2,   2,   2,   2
        dt        2,   2,   2,   2,   2,   2,   2,   1
        dt        1,   1,   1,   1,   1,   1,   1,   1

;==================================================================
        end
```

超 初心者向けJava入門

# 魁!! Dr.Jの JAVA-WALK ジャバウォック



## 第5回 ロジックを覚えよう！

### 腹立たしいコト限りなし！

やぁ，諸君！Dr.Dの従兄弟こと，Dr.J（ドクトルジェイ）である！世界では恐ろしくも悲しいことが起こっておる！我らが電脳界にゆかりの深い，とある米国メーカーのエグゼクティブも犠牲になってしまった！テロるテロればテロるとき！許すまじ民衆の敵！ワシの電脳による世界制覇の野望に立ちはだかる奸賊（かんぞく）めは我が怒りを知れい！パソコン神の鉄槌の第一撃はまずは電子メールによる抗議文って，何だかヘタレじゃのう！

まずはネットに接続して，…っと，おおぅ？チョット待てぃ，なんじゃこりゃぁぁ！？ナニ？readme.exe？ナントカ.emlって見覚えないぞい？あら，Nimdaってアンタ何ですか幸せですか今ってオイ！なんつーパワフリャアなウィルス！IISバックドアに添付ファイル，IEセキュリティホール，ありとあらゆる増殖手段を持つ史上最凶最悪のウィルス！ウムムム，これではまるでサイバー無差別テロではないか！？




既に世界で50万台以上が感染したとされるが，読者諸君は大丈夫じゃったかの？たぶんこの号が出ている頃ならばNimdaウィルスの猛威はだいぶん沈静化していると思うが，なにぶんしつこくしぶといタイプのウィルスである！まだしばらくは用心に越したことはないじゃろう！InternetExplorerのService Releaseとか，ウィルス対策ソフトの最新パターン定義ファイルとか，打てる手はすべて打っておくのだぞ！…いや，だから，ワシはホラ，他山の石って言うじゃろォ！？

### さて今月はロジックの基本

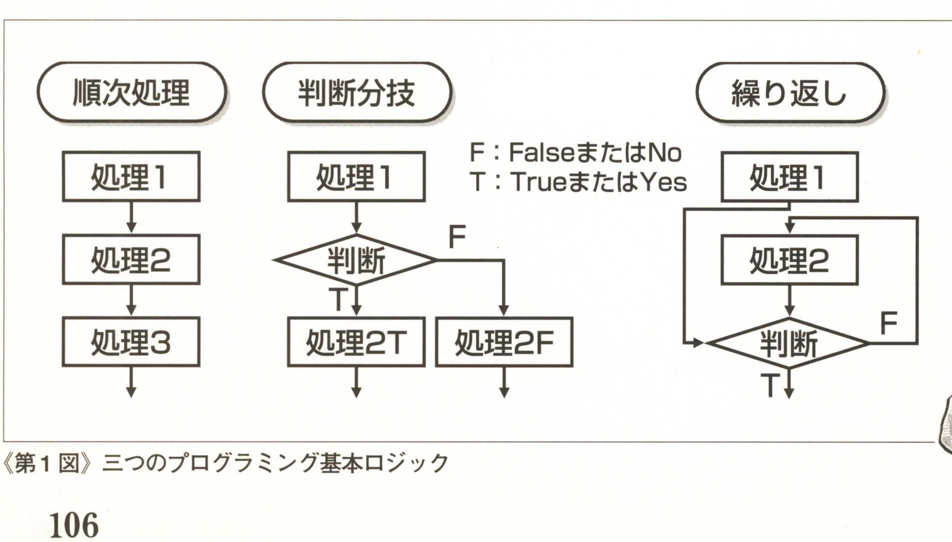
今までいくつかのJAVAの基本サンプル・プログラムなんぞを作ってきたが，応用範囲を広げることと，そもそものプログラミングの基礎を覚えるためにも，ロジックの基本をおさらいしておこう！

プログラミング・ロジックの基本は，「順次処理」「判断分岐」「繰り返し」の三つの動作である（**第1図**）。

JAVAの場合，判断のときには判断条件にあっていればTRUE(T)，ちがっていればFALSE(F)になる。論理定数というものを使うときがくるが，そのときにもこの真･偽の値を知っておくと便利なので，覚えておこう！

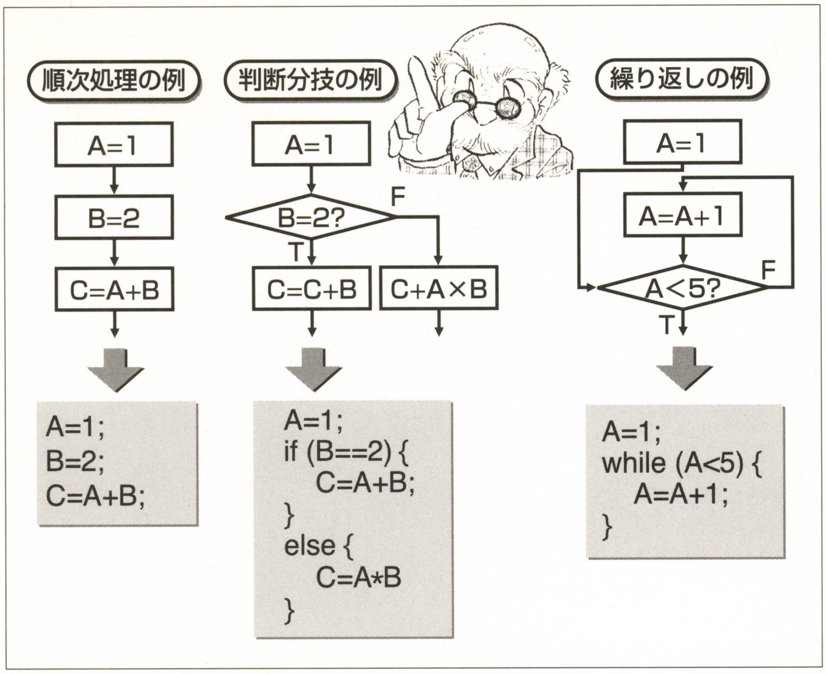
では，これらをJAVAで書くとどうなるか？これまでにいくつも例をやっているので，読者諸君には簡単すぎる話かな（**第2図**）。

どうかな？さて今回はじめて出てきたのがwhile文だ。whileは英語そのままに，～する間，という意味で，条件文が成立している間，中カッコでくくった中の処理を繰り返すというものだ。すなわち，中カッコ内の処理が条件文に反映されなければ，いつまでたっても処理が終了しないので，注意しよう！

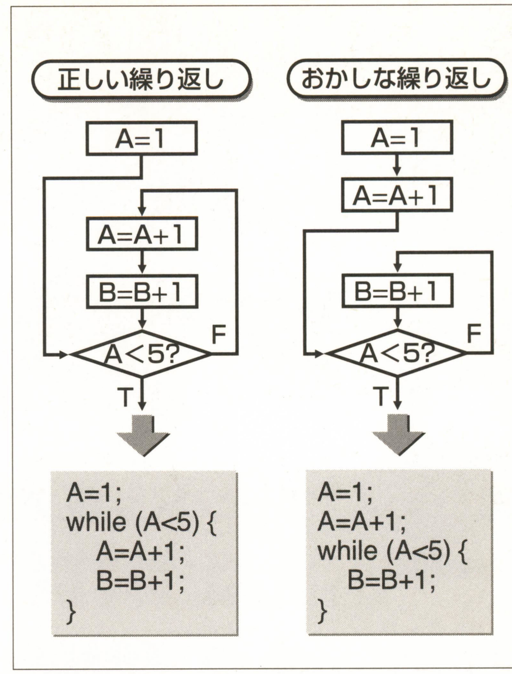



《第1図》三つのプログラミング基本ロジック






《第２図》三つの基本ロジックをJAVAで書くと…




《第３図》繰り返しの正しい処理とおかしな処理

第3図を見てもらいたい。こんな例はやりそうにないほど簡単じゃろうが，条件判断文がいくつもかさなってややこしくなってくると，ついつい誤ってしまいかねん。カッコがたくさんあると，わけがわからなくなってくるぞ！気をつけよう！

## まちがいだらけのプログラミング！

では，わかりにくい例を考えてみよう。たとえば，こういう例の場合じゃ。さてどこがおかしいかのう？

ファイル　Gucyazan.java

```
public class Gucyazan {
    public static void main(String args[]) {
        int a=1,b=1,c=1,d=1,e=1;
        for (i=0;i<10;i++){
            a=a+i
            if (a>5){
                d=d+c;
                while (d<10){
                    c=a+b;
                    while (e>1){
                        e=e+1;
                    }
                }
                e=e-d;
            }
            System.out.println("a=" + a);
            System.out.println("b=" + b);
            System.out.println("c=" + c);
            System.out.println("d=" + d);
            System.out.println("e=" + e);
        }
    }
}
```

ふっふっふっ，この中には大変なまちがいが一つと，まちがいだが，結果に影響がない（？）もの（それはフツーはバグというか？）と…コンパイラでもひっかかるかもしれないぞ。よーく考えてみよう！

さて，世界の終焉どころか身近なリストラ大流行と同時に秋風が吹きすさぶ，なんともやりきれん世の中であるが，せめて諸君ら次世代を担う若者達は，こんな時代だからこそ！JAVAプログラミングの一つも覚えて，明るい未来を築いていってほしい！

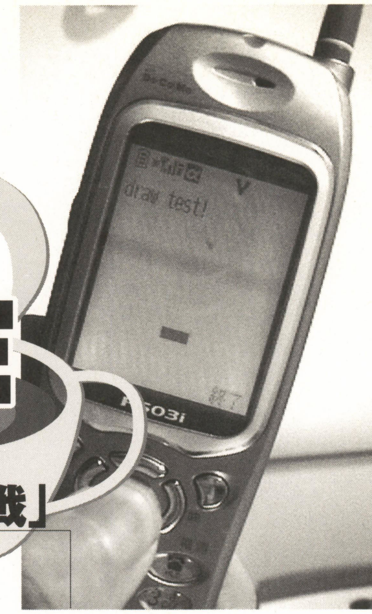
…というわけで，次回までに答えを見つけておいてくれたまえ。ちなみに，プログラミングには唯一の正解なんてものは存在しないので(まぁ唯一に限りなく近い答えというのは存在しうるが)，念のため！

ではまた来月，ワシといっしょに世界平和を取り戻そうではないか！



## 携帯電話で動くゲームを作ろう!!

# 君にもできる「iアプリ」制作



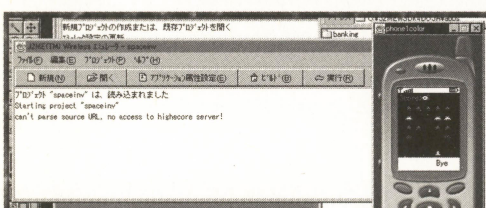
## 第8弾「フォントクラスを使ったｉアプリに挑戦」

（最終回）　いよいよこの連載も最終回です。iアプリの投稿作品はまだまだ募集し続けますのでよろしく。　（松原拓也）

## ｉアプリの作りかた

①サンマイクロシステムズのWEBサイト（http://java.sun.com/）からJava2の開発環境「Java 2 SDK Standard Edition 1.3.0」をダウンロードします。インストールすると“C:¥jdk1.3”というディレクトリにJavaコンパイラが展開されます。

② NTTドコモのWEBサイト（http://www.nttdocomo.co.jp/i/java.html）からｉアプリの開発ツール「J2ME Wireless SDK for the DoJa release2.2(以下，DoJa)」をダウンロードします。ダウンロードには名前の登録が必要です。インストールすると“C:¥J2MEWSDK4DOJA”フォルダに主要なファイルが展開されます。



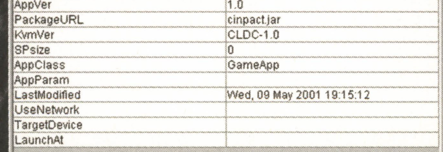
③スタートメニューから「J2ME～DoJa」を選択して，「KToolBar」を起動します。

④「新規」ボタンを押して，新規プロジェクト名を入力すると，“C:¥J2MEWSDK4DOJA¥apps”フォルダの下にプロジェクト名のフォルダと，その下に“bin”，“res”，“src”フォルダが作成されます。

⑤“res”フォルダに画像（gif）や音楽ファイル（mld），“src”フォルダにはソースファイル（java）を格納します。

⑥「アプリケーション属性設定」ボタンを押して，キーの値を入力します（“AppSize”と“Last Modified”は自動的に更新されます）。

AppName =（アプリケーション名）
AppVer = 1.0
AppClass =（最初に実行するクラス名）
SPsize =（スクラッチパッドのサイズ）
KvmVer = CLDC-1.0

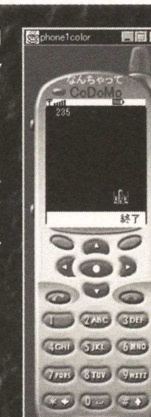


アプリケーション属性の設定 “cinpact”

| キー | 属性値 |
|---|---|
| AppName | チープインパクト |
| AppSize | 3559 |
| AppVer | 1.0 |
| PackageURL | cinpact.jar |
| KvmVer | CLDC-1.0 |
| SPsize | 0 |
| AppClass | GameApp |
| AppParam | |
| LastModified | Wed, 09 May 2001 19:15:12 |
| UseNetwork | |
| TargetDevice | |
| LaunchAt | |

⑦「ビルド」ボタンを押すと，コンパイルとプリベリファイが一括して行われ，jarファイルとjamファイルが“bin”フォルダに生成されます。jarファイルは"res"フォルダの中身とクラスファイルがzip形式で圧縮されたものです。

⑧エディタで“C:¥J2MEWSDK4DOJA¥lib¥devices¥phone1 color”フォルダの“phone1 color.properties”を以下のように書き換えます（値は120×130ピクセル時の設定です）。書き換えたら「編集一設定」をメニュー選択して，「デバイス指定」を“phone1 color”にします。



```
   :
screen.x=27
screen.y=65
   :
screen.width=120
screen.height=156
   :
icon.up: 44, 138, off
   :
icon.down: 44, 146, off
   :
screenPaintableRegion.width=120
screenPaintableRegion.height=130
   :
softbutton.0=0,138,36,18, softButton
softbutton.1=90,138,36,18, softButton
   :
```

⑨「実行」ボタンを押すと，携帯電話エミュレータ上でｉアプリが実行されます。

# フォントクラスを使ったｉアプリに挑戦

## 『銀河ウォーズ』

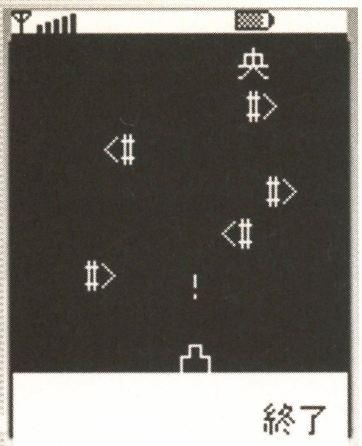
イメージ処理を使わずに，テキスト表示だけでゲームを作ってみました（**写真1**）。それと，今回初めてフォントのサイズ調整を行ってみました。

これで一応全機種の503iでまともに遊べると思います。

## 遊びかた

自機「凸」からミサイル「！」を発射して敵UFO「央」に当てて倒してください（**写真2，3**）。ミサイルの発射は決定キーです。ミサイルは発射後，左右のキーで誘導することができます（同時に自機も動きます）。そして上キーを押し続けるとミサイルが加速します。ミサイルが隕石「#>」に当たるとミスになります。ミサイルを3回隕石に当ててしまうか，時間切れになったら，ゲーム・オーバーです（**写真4**）。

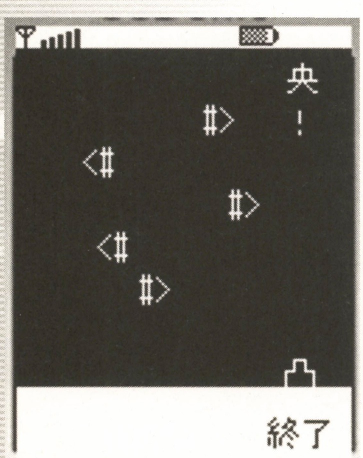
なお，このゲームはタイトーの「ギャラクシーウォーズ」をヒントにしています（あまり似てませんけど）。




◀《写真1》文字だけで作ったゲーム。イメージ処理を使っていないので，動きが速い




◀《写真2》ミサイルを誘導して画面上の敵ＵＦＯに当てろ！




◀《写真3》ミサイルが隕石に当たってしまうとミス




◀《写真4》3回ミスするか，持ち時間がなくなるとゲーム・オーバー

《リスト1》HTML ファイル“moji.html”

```
<HTML>
<HEAD>
<TITLE>i-Appli</TITLE>
</HEAD>
<BODY>
ｉアプリゲーム<BR>
<OBJECT declare id="b" data="moji.jam" type="application/x-jam">
MOJI</OBJECT><BR>
<A ijam="#b" href="moji.html">銀河ウォーズ</A>
<br>
</BODY>
</HTML>
```

《リスト2》JAM ファイル“moji.jam”

```
PackageURL = moji.jar
AppSize = 2641
AppName = 銀河ウォーズ
AppVer =
AppParam =
AppClass = GameApp
SPsize = 0
LastModified = Fri, 12 Oct 2001 04:27:52
KvmVer =
```

《リスト3》Java ファイル“GameApp.java”

```
/*-------------------------------------------------------------*/
/* iアプリクラス "GameApp.java" by 松原拓也*/

import com.nttdocomo.util.*;  /*クラスのインポート*/
import com.nttdocomo.ui.*;

public class GameApp extends IApplication
{
    public void start()
    {
        GameScreen screen = new GameScreen(); /*キャンバスの作成*/
        Display.setCurrent(screen);  /*キャンバスのカレント表示*/
    }
}
```

## 入力方法

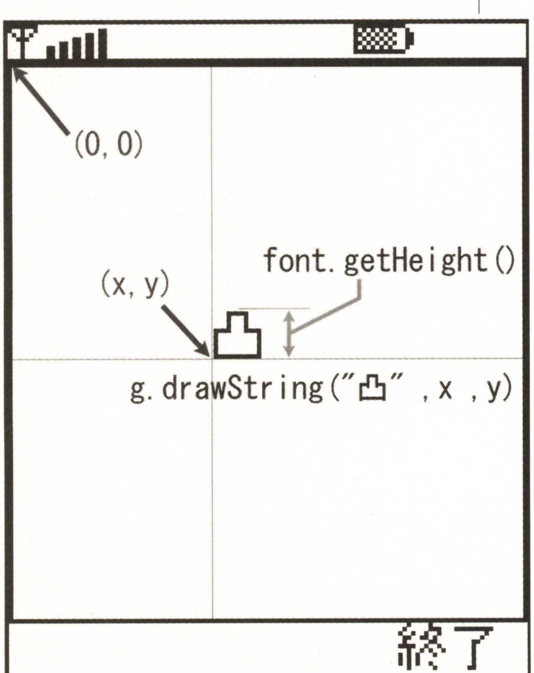
リスト1〜4をテキスト･エディタなどを使って入力します。あとは「iアプリの作成方法」を参考にしてビルドしてください。

## 技術的な説明

❶Stringクラスを配列にして，敵キャラクターの外見を格納します。今回は登場敵キャラクターを6個用意しました。そのうち1個を「敵UFO」に割り振っています。

❷ここで画面の左右に敵を配置します。ID番号が偶数か奇数かで敵の出現位置を分けました。さらに敵が1列に並んでしまわないように，座標と速度に乱数を付加しています。

❸文字列の描画は“g.drawString”メソッドを使います。このメソッドで描画される位置は，文字列の左下が基準となっています(**写真5**)。描かれる文字の大きさは，機種によってバラつきがあります（現状ではSO503iが14×14




▲《写真5》“g.drawString” メソッドの例。文字列の左下が基準となって描画される

《リスト4》Java ファイル “GameScreen.java”

```
/*-----------------------------------------------------------------*/
/* 「銀河ウォーズ」キャンバス "GameScreen.java"   by 松原拓也 */

/* クラスをインポート*/
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import javax.microedition.io.*;
import java.util.*;
import java.io.*;

public class GameScreen extends Canvas implements Runnable {

    final int MENW = 120;   /* 画面の幅・高さ[ピクセル] */
    final int MOJIW = 12;    /* 文字の幅・高さ[ピクセル]*/
    final int OBJ_MAX = 6;    /* 敵の数 */
    final int OBJ_UFO = 0;    /* 敵UFOのID番号 */
    final int SEQ_START=0;        /*シーケンス：スタート*/
    final int SEQ_READY=40;       /*シーケンス：READY表示*/
    final int SEQ_GO=45;          /*シーケンス：GO表示*/
    final int SEQ_GAMEOVER=1400; /*シーケンス：ゲームオーバー*/
    final int SEQ_END=1500;       /*シーケンス：終了*/

    Thread myThread;  /* スレッド */

    String [] strT = {"央","#>","<#","#>","<#","#>"}; /*敵キャラクター*/   ◀……… ❶

    int iTx[] = new int [OBJ_MAX];  /*被写体の座標*/
    int iTy[] = new int [OBJ_MAX];
    int iTx1[] = new int [OBJ_MAX];
    int iX , iY , iShoot;    /*ファインダーの座標、撮影用カウンタ*/
    int iScore;   /*スコア*/
    int iTama;
    int iSeq;     /*シーケンス*/
    Random rnd = new Random();   /* 新しい乱数列を作成*/

    /*-------敵または隕石の登場*/
    public void  objBone(int i) {
        if((i % 2)==0) { /*出現場所と移動量の設定*/
            iTx[i] = MENW + Math.abs(rnd.nextInt() % MOJIW);   ◀……… ❷
            iTx1[i] = - (Math.abs(rnd.nextInt() % 2)+1);
        }else{
            iTx[i] = 0 - Math.abs(rnd.nextInt() % MOJIW);
            iTx1[i] = Math.abs(rnd.nextInt() % 2)+1;
        }
        iTy[i] = ((i+1) * (MENW/8));
    }

    /*-------プレイデータのクリア*/
    public void objInit() {
        int i;
        for(i=0 ;i < OBJ_MAX ;i++) {
            objBone(i);
        }
        iX = MENW / 2;   /*自機の座標*/
        iY = MENW;
        iScore = 0;  /*スコア*/
        iTama = 3;   /*弾数*/
        iShoot = 0;  /*弾発射状態 (未発射) */
        iSeq = SEQ_START; /*シーケンスの初期化*/
    }

    public GameScreen() {    /*コンストラクタ*/
        setSoftLabel(Frame.SOFT_KEY_2, "終了");
        objInit();

        myThread = new Thread(this); /* スレッド作成 */
        myThread.start();  /* スレッド開始 */
    }

    public void paint(Graphics g) {  /*ペイントメソッド*/
        int i;
        int iHeight;
        Font font;
        String platform = System.getProperty("microedition.platform");

        if(platform.equals("SO503i") || platform.equals("D503i")) {   ◀……… ❸
            font = Font.getFont(Font.SIZE_SMALL);
        }else{
            font = Font.getDefaultFont();
        }

        g.setFont(font); /*フォントの設定*/   ◀……… ❹
        iHeight = font.getHeight();  /*文字の高さを取得*/

        g.lock();
        g.setColor(g.getColorOfRGB(0, 0, 0));  /*背景を描画*/
        g.fillRect(0, 0, MENW ,MENW  );

        g.setColor(g.getColorOfRGB(255, 255, 255));
        for(i=0 ;i<OBJ_MAX ;i++) {
            g.drawString(strT[i] , iTx[i] , iTy[i] );
        }
```

```
        g.drawString("凸" ,iX ,iY );  /*自機表示*/

        if(iShoot == 0) { /*ミサイル表示*/
            g.drawString("!" ,iX ,iY-iHeight);
        }else{
            g.drawString("!" ,iX ,iShoot);
        }

        g.setColor(g.getColorOfRGB(255, 64, 64));
        if(iSeq < SEQ_READY) {  /*テロップの表示*/          ◀··········· ❺
            g.drawString(" READY..." ,30 ,40 );
            g.drawString("Missile:"+iTama ,30 ,80 );
        }else if(iSeq < SEQ_GO){
            g.drawString("  ..GO!! " ,30 ,60 );
        }else if(iSeq < SEQ_GAMEOVER){
            /**/
        }else if(iSeq < SEQ_END){
            g.drawString("GAME OVER" ,30 ,40 );
            g.drawString("SCORE:"+iScore ,30 ,80 );
        }
        g.unlock(false);  /* ロック解除 */
    }

    /*----------*/
    public void run() {  /* スレッド実行 */
        int i;
        int iKey;

        for(;;) {
            try {
                Thread.sleep(50);   /*一定時間スリープ*/
            } catch(InterruptedException ignore) { }

            iKey=getKeypadState();
            if(iKey == (1 << Display.KEY_SOFT2))
                IApplication.getCurrentApp().terminate();

            for(i=0 ;i<OBJ_MAX ;i++) {  /*敵と隕石の移動*/
              iTx[i] += iTx1[i];

              if((iTx[i] < (-MOJIW))||(iTx[i] > (MENW + MOJIW))){
                  objBone(i);  /*画面外に出た場合の処理*/
              }
            }

            iSeq++;
            if(iSeq < SEQ_GAMEOVER) {
                if((iKey==(1 << Display.KEY_LEFT)) && (iX > 0))
                    iX-=4;    /*自機の移動*/

                if((iKey==(1 << Display.KEY_RIGHT)) && (iX < MENW))
                    iX+=4;    /*自機の移動*/

                if(iShoot == 0) {   /*ミサイル未発射なら、*/
                    if(iSeq > SEQ_GO) {
                        if(iKey==(1 << Display.KEY_SELECT)) {
                            iShoot = iY - MOJIW; /*ミサイル発射*/
                        }
                    }
                }else{ /*ミサイル発射中*/
                    iShoot--;
                    if(iKey==(1 << Display.KEY_UP)) {
                        iShoot--;  /*ミサイルの加速*/
                    }
                    for(i=0 ;i<OBJ_MAX ;i++) {
                        if((Math.abs(iTx[i] - iX)<MOJIW) && (Math.abs(iTy[i] - iShoot)<MOJIW)) {
                            iShoot=0; /*命中!!*/
                            objBone(i); /*敵消滅*/
                            if(i==OBJ_UFO) {  /*IDが敵UFOなら、*/     ◀··········· ❻
                                iScore+=100; /*得点プラス*/
                            }else{
                                iTama--;  /*隕石に当った場合、弾を減らす*/
                                iSeq = SEQ_START; /*シーケンス初期化*/
                            }
                            if(iTama==0) {   /*弾切れなら、*/
                                iSeq = SEQ_GAMEOVER; /*ゲームオーバー*/
                            }
                            break;
                        }
                    }
                }
            }else{
                if(iSeq > SEQ_END) {
                    objInit();  /*ゲームオーバー表示完了*/
                }
            }
            repaint(); // 再描画する。
        }
    }
}
```

ピクセル，D503iが16×16ピクセル，それ他の機種が12×12ピクセル）。そこで，“System.getProperty”メソッドを使って機種名を調べて，文字が大きい場合にのみフォント・サイズを変更します。フォント・サイズを変更するためには，“Font”クラスを使用します。フォントの大きさの種類は“SIZE_SMALL”“SIZE_MEDIUM”“SIZE_LARGE”の3種類です。

**（※フォント・サイズの機種依存については2001年10月号の「BLACK BOX」を参考にしてください）**

❹フォントの設定を行います。この命令以降で，フォントのサイズ変更が反映されます。

❺ここで，ゲーム開始前に「READY..」「...GO!」と表示させる処理を行ってます(先月号と同じ要領です)。こうした場面の状態は「iSeq」という変数で管理しています。

❻ミサイルと敵との当たり判定を行います。具体的には“Math.abs”メソッドで座標の差の絶対値を求めます。その値が一定量よりも小さいかで，接触したかどうかを判断します。

もし敵UFOに当たった場合はミサイルが消費せずに，得点が加算されます。隕石に当たった場合はミサイルが減り，弾数が0のときはゲーム・オーバー画面に移行します。

## 最後に

GIF画像を使っていないため，今までで最もファイル・サイズが小さい作品になりました。

今回のような文字表示のみのプログラムはF503iのような初期型の携帯でもキビキビと快適に動作するようです。絵が使えないと地味な画面になってしまいますが，ここは一つアイデア勝負でゲーム作りに挑んでみてはいかがでしょうか。

今月でこの連載は終了です。今までの内容を参考にして，おもしろいｉアプリを作ってみてください。投稿待ってます！



# 「BASIC STUDIO パワフルゲーム工房」入門

## 第8回 レースゲーム風デモを作る

文／山崎　寛

**今回はBASIC STUDIOの3D機能を使って，コース上をいくつかのキャラクターが動き回るデモを作ってみました。レースゲームなどを作る際の参考にしてください。**

『BASIC STUDIO パワフルゲーム工房』
発　　売：アートディンク
対応機種：PlayStation 2
メディア：CD-ROM 2枚組
　　　　　発売中
価　　格：9,800円（税抜）
http://www.artdink.co.jp/japanese/title/bs/index.html




### 加速，減速を表現する

レースゲームなどでキャラクターを動かす場合，左右で向きを変え，ボタンで加速，減速といった具合に，上下左右に直接操作するのとは違った動きをする場合が多いです。

このような動きを表現する場合，キャラクターの位置を示すX，Yの変数に加え，向いている方向を示す変数，動いている速度を示す変数の二つを用意します（第1図）。あとは，三角関数を使えば向き（角度）と速度からX，Yの移動量がわかるため，向きと速度を変化させれば自由にキャラクターを動かせるわけです。

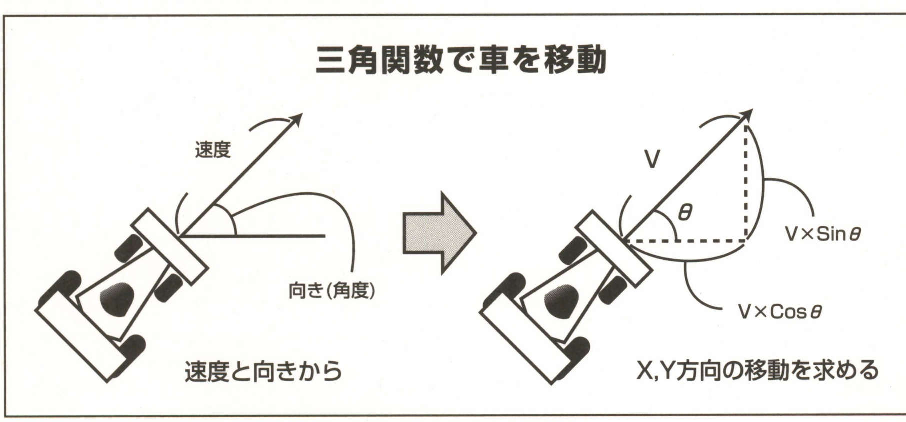
### CPUに道を教える

今回はデモということで，プレイヤーが自分で動かすのではなく，コンピュータがキャラクターを動かします。通常のゲームでもプレイヤー以外に複数のキャラクターが動く場合同様にコンピュータが動かすわけですが，この場合コンピュータが道の様子やどっちへ行けばいいのかを判断しなければなりません。

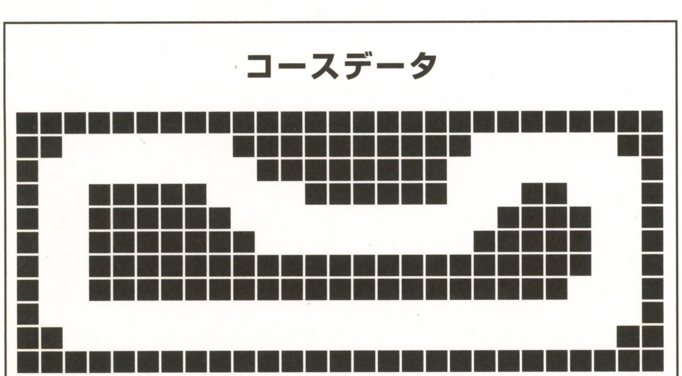
今回のプログラムでは，**第2図**のようにいくつかのブロックを並べたちょっとごつごつしたコースを走るのですが，画面には描かないもののコースデータに**第3図**のように走行する向きのデータを織りこんでいます。

ちょっとズルい手法ともいえるかもしれませんが，これでコンピュータのキャラクターは現在位置に書かれている向きの方向へと進んでいけばコースを周回できるわけです。

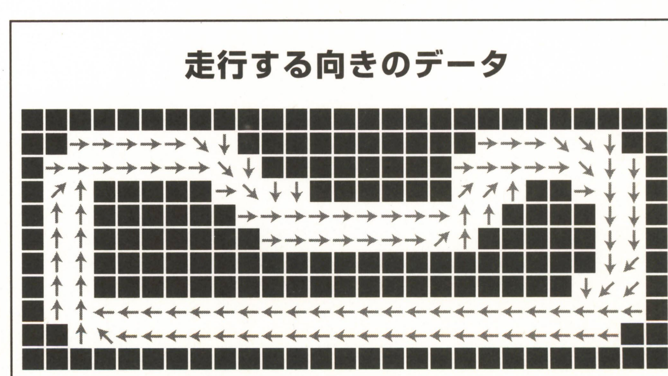
サンプルプログラムでの具体的な方法としては，コースデータに方向を数値で表したデータを持っておき，個々のキャラクターは現在の方向と比較し，同じな



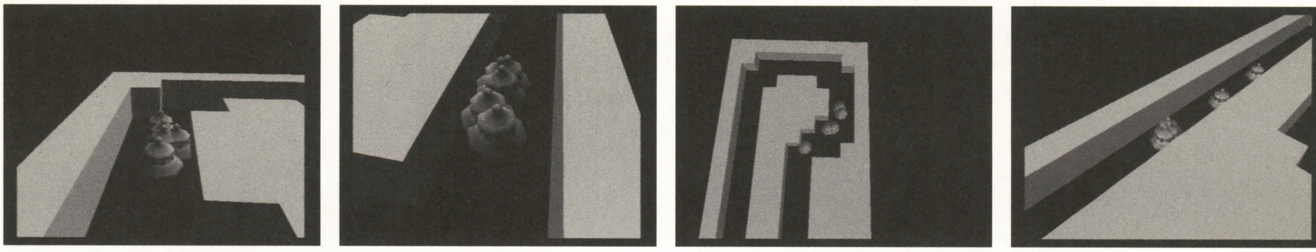

《第1図》速度と角度から移動させる




《第2図》今回のサンプルで使用しているコース




《第3図》コンピュータが迷わないように矢印をつける



《写真1》サンプル・プログラム実行画面

らば加速，違っていれば減速しつつその方向へと曲がるという形にしています。

## カメラを切りかえる

今回のプログラムでは，視点をコースの上からやキャラクターの後ろからではなく，あらかじめ決まったコース上のいくつかの点から選ぶという形にしています。

これは，あらかじめカメラを置く座標をいくつか決めておき，ターゲットとなるキャラクターにもっとも近い座標を選択しています。

カメラを決めたら，ターゲットとなるキャラクターを常に追いかけるような向きにカメラをセットします。これは，CAMVECという命令で，(カメラの座標－ターゲットの座標) と指定することで簡単に実現できます。

サンプルプログラムを実行すると，**写真1**のような画面で動きます。キャラクターは全部で5個動かしているのですが，それぞれの最高速度と加速をほんのちょっとずつ変えてあるため，同じプログラムで動かしているものの動きに少しずつ差がついていきます。

《リスト》レース風デモのプログラム

```
100 INIT
110 !●配列変数の宣言
120 DIM MAP(37,11) ―― コースデータ用
130 DIM TX(5),TY(5),TM(5),TS(5) ―― キャラクターのX座標，Y座標，向き，速度
140 DIM CX(4),CY(4),CZ(4) ―― カメラのX，Y，Z座標
150 !●コースデータ読みこみ
160 FOR I=1 TO 11
170   READ A$
180   FOR J=1 TO 27
190     B$=A$(J:J)
200     IF B$="A" THEN
210       MODDISP I*27+J
220       MODSET I*27+J,2,1
230       MODPOS I*27+J,J*36,1000,I*36
          } 210～230 コースデータがカベならば，カベの3Dモデルを表示
240       B$="-1"
250     END IF
260     MAP(J,I)=VAL(B$) ―― 配列にコースを入れる
270   NEXT J
280 NEXT I
290 !●動かすキャラクターの設定
300 FOR I=1 TO 5
310   MODDISP I ―― 3Dモデルを表示
320   MODSET I,8,8 ―― UFOを設定
330   TX(I)=6*36 ―― X座標
340   TY(I)=10*36 ―― Y座標
350   TM(I)=180 ―― 向き
360   TS(I)=0 ―― 速度
370 NEXT I
380 !●カメラの位置設定
390 CX(1)=16*36
400 CY(1)=10*36
410 CZ(1)=200
420 CX(2)=4*36
430 CY(2)=5*36
440 CZ(2)=800
450 CX(3)=16*36
460 CY(3)=3*36
470 CZ(3)=400
480 CX(4)=25*36
490 CY(4)=10*36
500 CZ(4)=900
    } 390～500 1～4の各カメラの位置を設定
510 CN=1 ―― 現在選んでいるカメラの番号
520 DO
530   FOR I=1 TO 5
540     M=MAP(TX(I)/36,TY(I)/36) ―― キャラクターの位置のコースデータを読む
550     M=MOD(M*45-TM(I)+360,360)
560     IF M=0 THEN
570       TS(I)=TS(I)+I*2 ―― 方向が同じなら加速
580     ELSEIF M>180 THEN
590       TS(I)=TS(I)-1
600       TM(I)=MOD(TM(I)-5,360)
610     ELSEIF M<180 THEN
620       TS(I)=TS(I)-1
630       TM(I)=MOD(TM(I)+5,360)
640     END IF
        } 580～640 方向が違ったら，その方向に曲げる
650     IF TS(I)>70-I*2 THEN TS(I)=70-I*2 ―― 最高速度の制限
660     IF TS(I)<0 THEN TS(I)=0 ―― 速度がマイナスにならないように
670     TX(I)=TX(I)+COS(RAD(TM(I)))*TS(I)/10
680     TY(I)=TY(I)+SIN(RAD(TM(I)))*TS(I)/10
        } 670～680 X，Yの移動量を計算
690     MODPOS I,TX(I),1000,TY(I)
700     MODROT I,0,RAD(360-TM(I)),0
        } 690～700 3Dモデルを移動
710   NEXT I
720   !
730   DI=SQR((CX(CN)-TX(1))^2+(CY(CN)-TY(1))^2) ―― カメラから目的キャラクターへの距離を求める
740   CAMPOS 1,CX(CN),CZ(CN),CY(CN) ―― カメラの位置を合わせる
750   CAMVEC 1,TX(1)-CX(CN),1000-CZ(CN),TY(1)-CY(CN) ―― カメラの向き設定
770   FOR K=1 TO 4
780     D2=SQR((CX(K)-TX(1))^2+(CY(K)-TY(1))^2)
790     IF DI>D2 THEN
800       CN=K
810       DI=D2 ―― 現在より近いカメラがあればそちらに切りかえる
820     END IF
830   NEXT K
840   FSYNC 1
850 LOOP
860 !●コースデータ
870 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAA
880 DATA AA0000012AAAAAAAAAA000112AA
890 DATA A000000012AAAAAAAA00000122A
900 DATA A76AAAAA0122AAAAAA776AA022A
910 DATA A66AAAAAA00000000066AAAA22A
920 DATA A66AAAAAAA000000056AAAAA22A
930 DATA A66AAAAAAAAAAAAAAAAAAAAA23A
940 DATA A66AAAAAAAAAAAAAAAAAAAA333A
950 DATA A664444444444444444444444A
960 DATA AA654444444444444444444444AA
970 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAA
    } 870～970 A…カベ 0…右 1…右下 2…下 3…左下 4…左 5…左上 6…上 7…右上
```

# Catlikeの もっと身近にプログラミング！

## 第5話 シューティング・ゲーム編　その4　敵弾アルゴリズム

**敵弾のないシューティングなんてシューティングじゃない！　なんて豪語したくなるくらい，敵弾は最重要。シューティング・ゲーム編の最後は，この敵弾について意地悪ぅく考えてみようと思うのです。**

### 筆者紹介

catlikeホームページ
http://catlike.marchen.to/
各種体験版がダウンロードできます。どうぞご利用ください～。

文:まかべひろし
最近モバイルはiPAQという手のひらサイズのコンピュータのみで行ってます。これにキーボードつけると,どこでもコンパクトにメールや文章が書けるマシンに早変わり!　これから手のひらコンピュータは来ますよ!　来年あたり,ずずずいっと。

絵:桜部のとす
intuosエアブラシが見つからない～!　タブレット用の特殊ペンなのですが,どこにしまったのやら,わからなく……アレがないと塗りにくくて困ります。でてこ～い。

### 風邪は万病のもとですよー

最近，風邪はやりすぎてませんか？　しかーも，なんか，今年の風邪はやたら長くありませんか？　ボクのゲーム開発仲間も，せーの，で一斉に風邪ひきさんになっちゃいまして，それでもゲームは作るから，開発部屋に風邪のウィルスが蔓延しているという恐ろしい状況になっているそーです。

それにしても，なんでこんなに風邪ひきさんが多いのかと思えば，今年の風邪は症状はそんなに重くないかわりに，なかなか治らないようです。だらだら，長々と風邪ひきさんであり続けなくてはいけないのです。

とはいえ，ボク自身は風邪なんて無関係の絶好調状態。元気の秘訣？　そりゃー，あったかくして，栄養あるもの食べて，毎日明るくいることですよー。

そこっ，「毎日プログラムで篭(こも)ってるからだろ」なんて，恐ろしい突っ込み入れないの！



### アルゴリズムに王道なし！ でも,人のコードは気になるもの

ではでは，シューティング・ゲーム編最終回。今回のシューティング・ゲーム編ではメインループを中心にお話してきましたので，ゲームのおおまかな組みかたに関してはわかっても，実際作りはじめていくと，「あのゲームのこの処理はどうやるんだっけ？」というような理想と現実の狭間(はざま)的な問題が出てくると思います。

そこで今回は，シューティング・ゲームで使いそうな処理に関して，項目別に紹介していこうと思うわけです。当たり前のように使われている処理でも実際組むとむずかしかったり，見様見真似で作ったために，ごちゃごちゃなプログラムになったりするかもしれません。まぁ，プログラムなんてものは正解はないので，動いてしまえばそれでいいのですが，それならこの処理，他の人はどーやる，というのを知りたくなるのは，プログラマーとして当たり前の心理でしょう。

今回紹介している方法以外に，自分はこーやる，という方法があったら，それはぜひCatlikeや他のベーマガ読者に教えてください。

### 弾よ！　主人公に向かって放たれよ！

まずは，シューティング・ゲームではぼ100％見られると言っても過言ではない，主人公に向けられる敵弾の処理についてです。

シューティング・ゲームのシチュエーションとしては，敵陣に単身突入する主人公，というのはよくある内容です。主人公は，相手の陣地に攻め込みに行くわけですが，敵側の立場からしてみれば，陣地に攻め込んでくる異物（主人公のことですね）を撃ち落とそうとするのは当たり前です。つまり，主人公に対して敵は弾を放ってくるわけで，まったく無関係の場所に向かって弾を撃つわけがありません。主人公機に向かってつぎつぎと襲ってくる敵弾をかわしていくのも，シューティング・ゲームの醍醐味と言える

でしょう。

で。この場合，敵弾の軌道というのは一直線になります。つまり，その軌道は敵弾がゲーム画面に放たれた瞬間に約束されているということになります。プログラム的に言えば，敵弾は，発射すると同時にその進行方向および軌道を決定しておかなければなりません。

ここで，以前，アクション・ゲーム編の記事で扱った「等速移動の算出式」が役に立ちます。

> 移動元座標（ax，ay）から，移動先座標（bx，by）に，s秒で移動する場合，移動開始からt秒経過したときの座標（x，y）。
>
> ```
> x = ( ax + ( bx - ax ) *s/t )
> y = ( ay + ( by - ay ) *s/t )
> ```

この式は，座標におけるa点とb点間をs時間で移動するとき，時間tのときは中間点内のどの座標になるかを算出するものです。となると，この式を使うために必要になる項目は，

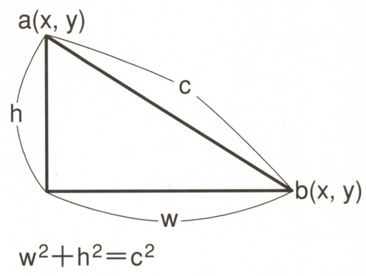
- **開始座標　a**
- **終点座標　b**
- **必要時間　s**
- **経過時間　t**

の四つとなります。ここでは，開始座標aを敵弾が発射される場所，つまり発射する敵の位置とし，終点座標bを主人公の座標としましょう。

ちなみに，以前この式を出したときは，2点間の移動として使いましたが，経過時間tが必要時間sを超えることで，終点座標bを突っ切って直進する，という移動に使うことができます。これは，敵弾の処理にぴったりですね。

さて，ここで困るのが，必要時間sです。普通に考えれば，主人公と敵の距離が短ければ少なく，長ければ大きくしなくてはいけません。もしどの弾も，発射されてから3秒で主人公に届くとなると，弾のスピードは距離によってばらばらになってしまいます。2点間の距離から，その速度を算出する必要があります。

では，距離はどのようにして求めましょう。これは，算数の問題。a点とb点を**第1図**のように，直角三角形として考えます。このようにすると，点aと点bを結ぶ直線は直角三角形の斜辺ということがわかります。この斜辺は，有名な「三平方の定理」で，長さを求めることができます。




**《第1図》三平方の定理**

点aの座標を（aX, aY），点bの座標を（bX, bY）とすると，必要な2辺の距離もABS（aX－bX）およびABS（aY－bY）で導き出すことができます。ちなみにABSというのは，絶対値のこと。BASICやC言語で用意されている関数のひとつです。求めたいのは直角三角形のwとhの長さ。座標は点aと点bでどちらが大きい数値かわからないので，理論上は，絶対値を求めてやる必要があるわけです。

そして，三平方の定理に基づいて，2辺の長さをそれぞれ2乗してから足します。で，その結果の平方根を求めればいいのです。

プログラムで書けば，以下のようになります。

```
c = sqrt ( pow ( ( aX - bX ), 2)
+ pow ( ( aY - bY ), 2 ) );
```
※pow( )はべき乗計算を行う関数，sqrt( )は平方根を求める関数。

説明のうえでは，絶対値を得る必要があると言いましたが，aX－bXの値が正であれ負であれ，2乗することで必ず正の値になります。ですから実際の式では絶対値は求めていません。

これで距離cが求まりました。距離cから必要時間sを求めるには，距離cをスピードで割ってやればよいのです。たとえば，1フレームに弾は3進むとしたら，距離c÷3で必要時間sを求めることができます。

最後の必要項である経過時間tは，発射した瞬間を1とし，ゲーム内時間が進むたびに増やしていけばよいことになります。

これで，主人公機に向かってまっすぐに飛んでくる弾ができました。このような弾を取り入れることで，プレイヤーは常に動き回らざるを得なくなり，とても緊迫したゲーム展開が期待できます。

## まっすぐの敵弾，もうひとつのアプローチ

さて，すでに，主人公に向かってくる弾の処理というのは完成したわけですが，ここでもうひとつの方法を提示しておきたいと思います。かなり，かな〜り昔のシューティング・ゲームをご存じの方は，敵弾に関するフレーズに，このような売り文句を聞いたことがあるかなと思います。「弾の精度はなんと32方向！」これはいったい何なのかというと，昔のシューティング・ゲームは弾の発射される角度が何種類かに決まっていたのです。それこそ4方向とかというと興ざめですが，この売り文句のように32方向ともなると，結構自然に，自分に向かって撃たれているような感じがします。

ちなみに，さきほど紹介した，2点間での処理を行う場合，この角度数は理論的に360種類以上です。昔は方向数を売りにしていたくらいなのだから，360以上の方向を算出できるならそれでいいじゃないか，と思うかもしれません。しかし，実際プレイすると，これでは難易度が高くなってしまうのです。必ず自分に向かってくる敵弾，それは，まったく休む暇がないことを示します。

もうひとつ，角度数を決めて作る理由があります。シューティング・ゲームには，安全地帯と呼ばれる，絶対に弾も敵も飛んでこない場所があります。安全地帯を設置することで，ある程度難易度を下げ，かつ「ゲームの攻略」というものを現実的にします。この，安全地帯を作るためには，必ず主人公に向かってくる弾は使うことができません。精度の限界がある，角度数の決まった弾を使うことが重要です。

たとえば，上下左右4方向にしか撃たれない弾の場合，斜め方向が安全地帯であることは誰の目にもあきらかでしょう。これが32方向になったとしても，敵弾がある程度遠くからしか撃たれなければ，絶対に弾の当たらない部分というのが生まれます。とくに，ボス戦など攻略

性が高いシーンに使うのが適していると言えます。

長々とその必要性を説明してきましたが，プログラムのほうは至って簡単。角度というのは360度ですから，それを32で割って，その方向に進む場合，1フレームで進めるX，Y成分を用意すればいいわけです。おなじみ，テーブルの考えですね。

```
MoveTbl [32] [2] ={
  {0, -8}, // 0（真上）
  {1, -6}, // 1
  {2, -4}, // 2
  :
  :
};
```

こんな感じ。テーブル内の数値はゲーム性によって調整してみてください。あとは，弾を発射する敵と主人公の位置から，このテーブルのどれを使うかを指定するだけです。もちろん，まっすぐにしか弾を飛ばさない敵や，このテーブルを利用して32方向に，花火のように同時に弾を発射するボス，みたいな敵がいてもいいでしょう。

重要なのは，このテーブルを使うことによって，ゲームの難易度を調整することができる安全地帯を生み出し，ゲーム性を高めることができる，という点なのです。

## 誘導弾を考える

まっすぐに飛びつづける弾はできました。基本的に弾というのはまっすぐに飛びつづけるものです。しかしそれでは，ほんの少しの手元でのずれが，距離に比例して精度を落とすということにつながってしまいます（その法則を利用しての「安全地帯」でもあるのですが）。

ゲームの世界ではそれでも良かったのかもしれません。しかし，シューティング・ゲームという娯楽が生まれる以前には，弾というのは「確実に標的を撃ち落とす」ために使われていたので，それでは問題がありました。そこで人間は，命中率をより高めるため，弾自身が標的を認識して追いつづける，誘導弾というものを作りだしました。もちろん，戦争のために…。

悲しいかな，ゲームというのは現実の模倣から生まれました。現実に取り入れられた武器がゲーム内に出ないわけはありません。シューティング・ゲームにおいて，誘導弾も日常茶飯事に使われる敵弾処理のひとつです。これが放たれたが最後，主人公は誘導弾をどうにかしてかわすために大きな苦労を負うことになるのです。

さておき，ゲーム内でこの誘導弾を出すとなると，いろいろな方法を考えることでしょう。まずは，多くのゲームで取り入れられている誘導弾の動きを思いだしてみます。誘導弾は画面内に放たれるとまず，主人公側に向かって飛んできます。ここまでは，既存の「まっすぐ弾」と同じですが，主人公がそれをかわそうと移動すると，その移動した方向に合わせて，あろうことか誘導弾はその進行角度を変え，さらに主人公に追いすがってくるのです。

ということは，考えられるのは，主人公が動いたらすぐに，主人公との位置を確認し，移動方向の補正を行うのでしょうか。いえいえ，主人公が動いてすぐにそれを行っているのなら，主人公は誘導弾を避けることができません。

ということは，誘導弾がその進行角度を変更するのは，一定時間に一度，ということになります。毎フレーム変更するのではなく，60フレームに一度とかなら，その隙にかわすことができるかもしれません。

じゃあ，まっすぐ弾の経過時間tを参照しつつ，60フレームに一度，進行方向を変えるようにプログラムをしてみましょう。要は，そのたびに進行方向の再計算を行うだけです。使うアルゴリズムは，2点間の移動計算を使ってみてください。うまくいったように見えますが，何度かテストプレイを行うと，異様な光景を目にすることがあるはず。

イメージしてみましょう。縦スクロール・シューティング・ゲームで，画面上部から放たれた誘導弾をぎりぎりでかわし，60フレームのうちに誘導弾の逆方向に回りこんでみてください。すると，今まで下方向に進んでいた誘導弾が，いきなり180度方向を変えて，上方向に向かうではありませんか。誘導しているのだからこれで問題ないと思えますが，いくらなんでも，今までの移動エネルギーを完全に中和して180度逆に向かうのは，物理法則を無視しています。それに，こんな動きをされたのでは，プレイヤーも誘導弾をかわしきれずにダメージを受けてしまうでしょう。もっと，こう，緩やかなカーブを描きつつ方向を変更させてやりたいものです。

もちろん，今まで60フレーム毎の変化だったのを，30フレーム毎にするなど，間隔を狭めてやれば自然な動きになります。しかし，あまりに短い間隔だと弾を避けるのが異様にむずかしくなったりと，どっちつかずな気もします。

そこで，まっすぐ弾アルゴリズムでも，精度の低い，角度テーブルを使って考えてみることにします。問題点は，回りこんだ際に急激に曲がることなのですから，曲がることのできる角度に制限をつけてしまえばいいのです。角度テーブルを用いる以上，今，テーブルのどこを参照しているか，というのを保持しているはずです。そこを中心に左右45度までを，1回の再計算で曲がる限界値にしてしまいましょう。再計算によって，つぎに曲がる角度が前回から45度以上離れていたら，45度の位置までに補正してしまうのです。

すると，180度曲がるには4回の再計算を要することになります。これは結構自然な曲がりかたに見えますし，プレイヤーも問題なくかわすことができるのではないでしょうか。

また，誘導弾とも言えど，永遠に追いつづけられては，いつかはプレイヤーは死ぬ運命となってしまいます。ある程度誘導したら，いさぎよく誘導弾は消えてしまわなくてはなりません。では，その判定はどうするのでしょう。方法として，再計算の回数を決めておく，画面外に一度出たら再計算しない，最初の発射角度から180度以上は曲がらない，等を挙げておきます。

一番簡単なのは再計算の回数を決めておくことでしょうが，一緒に，画面外に出た場合も消えるようにしておくのも良いかもしれません。するとプレイヤーは誘導弾を早めに消そうと画面外に誘導弾

を誘導しますから，そこで画面外から現れる敵とはちあわせ，という状況を作りだすことができます。

ここまでくれば，あとは，誘導弾の精度を，何度かテストプレイして決めることになります。ポイントは，何フレームに一度再計算を行うか，そして1回の再計算で曲がることのできる角度は最大何度までとするか。何度まで再計算を行うか。そして，説明しませんでしたが，誘導弾の移動速度。この4点の数値を，自分のゲーム上で納得できるまで調整してみましょう。

多くのゲームでは，誘導弾は他の敵弾よりも目に見えて遅いスピードで移動するようにしています。誘導弾はプレイヤーにとって非常に厄介な相手のひとつです。スピードを落とすことにより，難易度を落としているのかもしれません。まぁ，誘導弾というものがゲームで使われた初期は，その計算精度の粗さを隠すために速度を遅くしたという話もありますが…。それが伝統となって，誘導弾はスピードが遅いもの，となったのかもしれませんね。

## 「金魚のしっぽ」子機にはバッファで対応

最後に，パワーアップ型シューティングに見られる，子機についての考察をしましょう。どういうものかというと，アイテムを取ると主人公の横に現れて，主人公と一緒にミサイルを撃ったり，敵弾をガードしたりしつつ，主人公の動きを後追いするように移動するかわいいやつです。名作シューティングの『グラディウス』では「オプション」と呼ばれる発光物体でしたので，オプションと呼んだほうがわかりやすいかもしれませんね。では，主人公の後を，金魚のしっぽのごとく追いまわすアルゴリズムはどうなっているのかを考えましょう。

あ，金魚のしっぽって，ボクの造語です。ホントの言いかたはちがうのだけれど，ちょっと汚いしね。金魚にしっぽはない？　まぁまぁ，意味はなんとなーくわかるでしょ。ずーっと主人公の後をついてくる子機君にはぴったりの言葉だと思うんですけどね。

で，このアルゴリズム。実際のゲームの子機の動きをじーっと見ていればすぐにわかるでしょう。そう，子機は，主人公の「以前何フレームか前にいた場所にいる」のです。ということは，プログラム的には，「主人公が以前いた位置を保存しておく」ことになります。

ここまで気づいてしまえば後は簡単な話。子機は主人公より60フレーム遅れた位置で動くとしたら，60フレーム分，主人公の位置を保存しておく場所が必要になります。座標はX，Y座標で表せますから，posbuf［60］［2］のような配列を用意することで実現できます。

この配列は，タイピング・ゲーム編で紹介しました，リングバッファとして用います。つまり，書き込みを続けて一番最後まで行ったら，最初に戻って60フレーム前のデータに上書きするように書き込みを続けるという形です。この書き込みと，書き込みポインタの移動処理は以下のようになります。

```
//配列とポインタの設定
int posbuf [60] [2] ;int
pospnt = 0;

//書き込み処理。x, yを書き込む
posbuf [pospnt] [0] = x;
posbuf [pospnt] [1] = y;
pospnt = ( pospnt + 1 ) %60;
```

で，子機はこの配列のどこを参照すれば良いのかというと，posbuf［(pospnt－59）%60］となります。こうすることで，60フレーム前の主人公の位置を，子機の位置として使うことができるのです。

問題は，この書き込みルーチンを実行するタイミング。しつこく「60フレーム前」と書いてきましたが，フレームという言葉に，素直に毎フレーム書き込み処理を行うと，子機はイメージとちがう動きをするかもしれません。

毎フレーム書き込むと，子機は，主人公が移動していようがいまいが，きっかり60フレーム前の主人公の位置を模倣し，主人公が60フレーム間移動しないで止まっていれば，主人公と子機は重なってしまうことになります。

こういう子機の動きもアリですが，主人公が移動していないときは子機にも止まっていてほしいと考える人もいるでしょう。その場合，書き込みルーチンを実行するタイミングは，毎フレームではなく，主人公が移動したときとなります。こうすることで，子機は，主人公の60コマンド前の位置に来るようになります。

どちらの子機の動きも魅力的ですし，ゲーム性にも大きくかかわってきます。ゲーム・デザインをするときに，どちらがいいか，自分で決めてみてください。中には，そう，最初に子機の移動タイプを選択できるゲームなんていうのもありましたね。

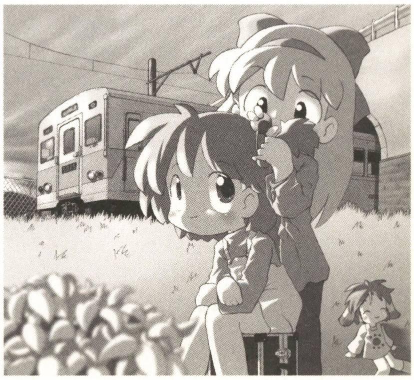
## 後はさぁ,作るのみだね!

というわけで，シューティング・ゲーム編いかがでしたか？　メインループというゲームすべての基本を組むところからはじまって，シューティング・ゲームには欠かせない，敵側を制御するアルゴリズム，そして今回の，敵弾の処理。これらを組み合わせるだけでも，結構立派なシューティング・ゲームを作り上げることができるでしょう。

ですが，一番大事なのは，出来上がってからの微調整。シューティング・ゲームはとくに，プレイヤーの反射神経や腕，そして動作環境を問うジャンルです。多くのテスト・プレイヤーに参加してもらい，どんな人でも楽しめるゲームを目指して，がんばってみてください。

そうそう，最後に，シューティング・ゲームを大量生産してる友人，Studio Sequenceさんのお言葉。
「プログラマーが簡単だって思うくらいの難易度が一番ちょうどいいんだ」

プログラマーは作ってる間ずっとこのゲームに触ってるわけですから，なるほど，納得できる言葉です。



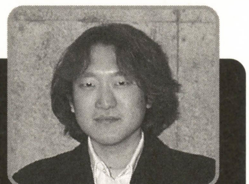
# DTM Art Studio
# パソコンで音楽をたのしもう!

協 力
ヤマハ（http://www.yamaha.co.jp）

Text&SoundData
仲野方敏／上村学

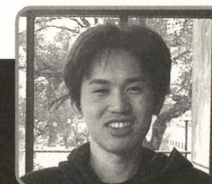
**このコーナーは，芸大生が体験したDTMの基本，応用，実践を紹介するコーナーです**

## 筆者紹介



**仲野方敏：**
大阪芸大
芸術学部　音楽学科
音楽工学コース（音楽工学専攻）3回生

こんにちは。「らくらく作曲名人2」を使っての連載は今回が最後です。次回からは，本格的なシーケンス・ソフトを使っての連載がはじまります。今回の記事の内容も含めて，復習の意味でも，とにかく曲を書き続けましょう。そうすることで，確実にステップ・アップしていきます。それでは，今回もがんばっていきましょう。



**上村学：**
大阪芸大
芸術学部　音楽学科
音楽制作コース（作曲専攻）3回生

みなさん「らくらく作曲名人2」の使いかたにはもう慣れましたか？この「らくらく作曲2」のソフトの連載も今回が最後です。このソフトはアレンジに強く，しかも簡単にアレンジができるので，楽しかったですね。さあ，それでは今回もはりきって音楽制作をはじめましょう。

# YAMAHA「らくらく作曲名人2」
## ～作曲編～

### 先月号のおさらい

前回はさらに進んだアレンジに挑戦ということで，より音楽的なアレンジの方法を説明しました。いくつかのジャンルを取り上げて，それぞれのジャンルの音色やリズム，コード進行について説明，さらにらくらく作曲名人2を使ってそれらのジャンルにアレンジするときのコツやヒントも説明しました。また，私達芸大生ならこんな方法でアレンジをする，という一例も紹介しました。

今回は，この「らくらく作曲名人2」の連載最後の総仕上げとして，作曲からアレンジまでやってしまいたいと思います。そして，その他にこの「らくらく作曲名人2」の今までに説明しきれていない便利な機能の使いかたも説明したいと思います。

これであなたもらくらく作曲名人2をマスターかな？

### 「らくらく作曲名人2」

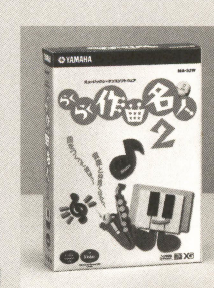
「らくらく作曲名人2」は，音楽の専門知識がなくても，作曲・編曲等が簡単にできるるソフトです。「めちゃらく作曲名人」とちがい，シーケンサとして使うこともできるので，DTM初心者以外にもお勧めです。それに，楽譜を制作することも可能なので，多くの人が楽しめるソフトです。

**製品仕様**

- CD-ROM（「らくらく作曲名人2」，ソフトシンセサイザ「S-YXG50」，ヤマハCBXドライバ，MidRadioプレーヤ，用語集）
- 取扱説明書

**動作環境**

- 対応パソコン：NEC PC-98シリーズ，IBM-PC／AT並びにその完全な互換機
- 対応OS：WindowsMe/98/95/NT4.0/2000
- CPU：Pentium-166MHz以上/Celeron以上（ソフトシンセサイザ同時使用の場合MMX Pentium-233MHz／Celeron以上推奨，Windows2000の場合PentiumⅡ300MHz以上推奨）
- 必要メモリ：32MB以上（64MB以上推奨）
- ハードディスク必要容量：40MB以上
- ディスプレイ：800×600ドット，24ビットTrue Color以上
- 音源：ソフトシンセサイザまたはXG対応MIDI音源機器
- 必要機器：マウス，ハードディスク，2倍速以上のCD-ROMドライブ



14,800円

### メロポンボタンで作曲しよう

さあ今回はアレンジだけにとどまらずメロディも作曲していきましょう。

このらくらく作曲名人2にもめちゃらく作曲名人で使ったメロポンボタンがあります。このメロポン機能を使うと，らくらく作曲名人2でも，めちゃらく作曲名人のようにゲーム感覚で曲作りが

できてしまいます。

### メロポンテンプレートの選択

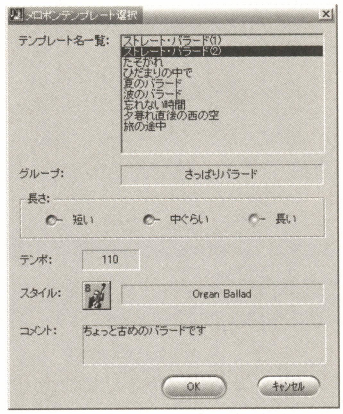
メロポン機能を使って作曲するには，まずステージ画面でステージの右下に表示されているコンダクタ（指揮者）にマウスポインタを合わせるか，［表示］メニューの中から［メロポンパレット］を選んで，メロポンパレットを開きます（写真1）。




《写真1》メロポンパレットを開く

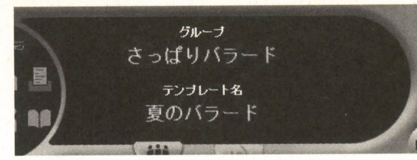
つぎに，このメロポンパレットから作りたい曲のイメージやスタイルを決めるためのテンプレートを選びます。たくさんの種類のテンプレートがパレットの中の各アイコンで表されています。パレットの右端にある二つの三角形のボタンで，パレットのつぎのページや前のページを表示させることができます。パレットの中のアイコンをクリックするとテンプレートが選択され，ステージ上に選んだテンプレートのプレーヤーが配置されます。

また，選んだテンプレートで使われているスタイル名がステージの左側のカーテンに表示されます。さらに細かくテンプレートを設定するには，パレットの中のアイコンをマウスの右ボタンでクリックして，［メロポンテンプレート選択］を選んでください（写真2）。




《写真2》［メロポンテンプレート選択］画面

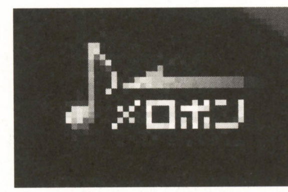
メロポンテンプレート選択画面では，テンプレートアイコンのジャンルの中からさらに細かい曲調を選択できます。また曲の長さも三つの中から選ぶことができます。それぞれの曲調にコメントやテンポ等も表示してくれるので，参考にして選ぶとよいでしょう。また，テンプレートを選択すると，選んだテンプレートの属するグループ名とテンプレート名が，ステージ画面の中央下にあるコンダクターパレットに表示されます（写真3）。




《写真3》テンプレートの属するグループ名とテンプレート名が，コンダクターパレットに表示

### メロディを作る

さあここでやっと登場のメロポンボタンです。メロディを作るには，画面左下の［メロポン］ボタンをクリックします（写真4）。




《写真4》画面左下の［メロポン］ボタン

すると，さっき選んだメロポンテンプレートに合ったメロディがランダムに作曲されます。［再生］ボタンをクリックして，できあがったメロディを確認してみましょう。もし，メロディが気に入らなかった場合は，もう一度メロポンボタンをクリックして，気に入るまで何度でもメロディを作り直すこ

## 民族音楽のスタイル

### 沖縄の音楽

沖縄の民族音楽も日本の民族音楽としての認識があると思います。しかし，沖縄は1609年に薩摩に制圧されるまでは，琉球国という国であって，独自の文化が根付いていました。琉球国は，国の位置条件から東南アジアや中国，日本等との貿易が盛んで，それらの国の文化の影響を受け，いろいろな国の文化が混ざった独自の文化が発展しました。

たとえば，沖縄の音楽を語るうえで欠かせない楽器として「三線」（さんしん）という楽器があります。これは，もともとは中国から伝わってきた楽器なのです。日本にも「三味線」（しゃみせん）という楽器がありますが，これは沖縄から日本に伝わったという説があります。

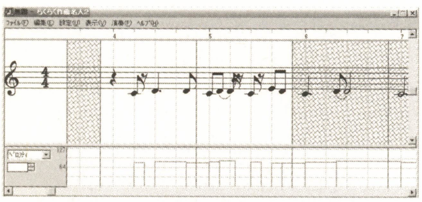
三線と三味線は共通点もありますが，一番のちがいは，三味線が猫や犬の皮を張っているのに対し，三線は蛇の皮を張っている点です。三線に張ってある蛇の皮は東南アジアから輸入した，にしき蛇の皮を使っていて，昔からの貿易の影響が現れた，まさに沖縄を象徴する楽器だと思います。それから楽器の長さもちがい，津軽三味線は約94cm，琉球三線は約77cm，唐三線は約91cmです。

そして，沖縄音楽の一番の特徴は何といっても音階にあります。その音階とは「ド・ミ・ファ・ソ・シ・ド」の琉球音階といわれるもので，沖縄独特の要素です。これはインドネシアのペロッグ音階と呼ばれる音階とほとんど同じで，似たような要素はポリネシアや東南アジアや中国南部やヒマラヤ地方にも散在しています。また，エイサーやカチャーシーといわれる自由で即興的な乱舞などに見られる，裏打ちのきいたリズム感も，日本では沖縄だけの独特な要素です。

このような要素を用いた沖縄民謡は，歌三線といわれ，歌が主役で，三線は脇役の伴奏という形式があり，三線の独奏曲はほとんどありません。また沖縄民謡には，三線の簡単な音高を漢字で縦書きにし，横に歌詞が書いてある「工工四」（くんくんしー）という楽譜があるだけで，細かい表現や歌の音高はほとんど書かれてなく，沖縄民謡を勉強するには師匠について手習いで勉強していくのがほとんどです。縦書きの楽譜を使うところは日本の琴や尺八等の楽譜と共通するところです。

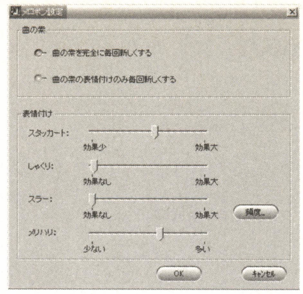
このように，いろんな国の古い文化が混ざった文化に，私自身大変興味があって1年半ぐらい前から三線を習いはじめてしまいました。琉球音階の魅力にも吸いよせられ，琉球音階を使った曲まで書きはじめています。先月号で紹介した“～風のアレンジ”みたいですが，みなさんもらくらく作曲名人2で琉球音階のメロディに伴奏を付けてみてはいかがでしょう。琉球音階を使うだけで，沖縄風の曲が作れてしまいます。

とができます。また，メロポンボタンでは，最初の2小節の曲の素（モチーフ）の作成と，その後の展開を作曲しますが，楽譜画面を開いて最初の2小節の斜線が入っていない部分は自分で編集することができます。メロポンで作った曲の素（モチーフ）の気に入らない部分を変えてもいいし，動機（モチーフ）をまったく新しく自分で作ることもできます（**写真5**）。

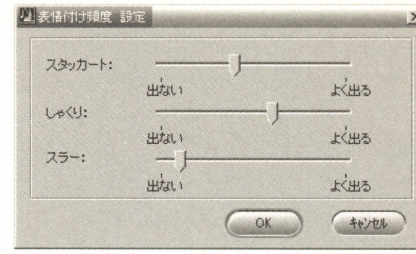


《**写真5**》楽譜画面で編集，新規作成

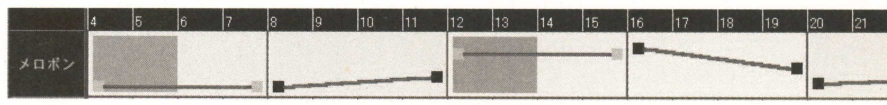
また，その際，［設定］メニューの［メロポン設定］からメロポン設定画面を開き，［曲の素の表情付けのみ毎回新しくする］を選んでください。そうしないとせっかく作った2小節の曲の素（モチーフ）まで新しく作曲されてしまいます。この，メロポン設定画面では，曲の素の作曲設定の他にも，表情付け（曲の素を展開させていく部分）の設定もできます。スタッカートやしゃくり，スラー，メリハリを設定できます（**写真6**）。



《**写真6**》メロポン設定画面で詳細に設定




《**写真7**》表情付けが生成される頻度の設定




《**写真8**》アレンジ画面のメロポン表示領域

また，［頻度］をクリックすると，ここで設定した表情付けの生成される頻度を設定できます（**写真7**）。

## メロディの編集

ステージ画面で足下が赤色のプレーヤーをダブルクリックしてアレンジ画面を表示すると，メロポンボタンで作ったメロディを編集することができます。メロポンボタンで作ったメロディをさらに自分の思いどおりに盛り上がりを付けたり，部分的に作り直すことができます。

アレンジ画面の一番上段のメロポンと書いてあるメロポン表示領域に表示されている直線の端の黒い■をドラッグして上下させることによって，メロディの盛り上がりを変更することができます。

ただし，グレーの■はドラッグできません（**写真8**）。

また，メロポン表示領域のメロポン・アイコンは，4小節単位の楽節をイメージ化したようなもので，ブルーのメロポン・アイコンはサビ以外の部分に置かれ，黄色のメロポン・アイコンはサビの部分に置かれています。このように，わけられたメロディの範囲を

## シリーズ音響学

シリーズ音響学最終回の今回は，前回の最後でも少しだけ説明した聴覚の諸効果に焦点を絞ってみます。

私たちの聴覚には特有の効果というものがあります。ミキシングなどをやるうえで，こういったことに関する知識が問題解決の糸口になる場合もあります。ですので，みなさん，ぜひしっかり勉強してください。

### 立体音（ステレオ）

両耳を使って音を聴くと，片方の耳だけを使って聴いた場合より音源の方向を感知したり，空間の広がりや遠近感が得られるといった利点があります。この効果は，音の「定位（位置）」に関して最もはっきり現れます。両耳で聴くことによって，私たちはかなり正確に音の位置を把握することができます。試しに，目を閉じたまま誰かに鈴などを鳴らしてもらうとわかると思いますが，ほんの少し音源が横に移動しただけでも，私たちはそれを感じ取ることができます。

ちなみに，音源の横移動はかなり正確に感じ取れますが，それに比べて上下の音源移動に対しては，かなり鈍感になります。二つの耳によって，空間の音を立体音としてとらえている人間の聴覚の特性を利用して，二つ以上のチャンネルを使い，できるだけ原音場（録音された音が本来存在している空間）と同じような立体音場を再現するために作られたのがステレオです。

なお，ヘッドフォンを使った音場の再現は，耳に音が到達するまでの空間のクロストーク（左チャンネルの音と右チャンネルの音が耳に到達するまでの空間の中で混ざり合うこと）がないので，音場のイメージを頭の中で再現する点において，スピーカ再生とは大きく異なります。これを「両耳効果（バイノーラル）」といいます。

### ハース効果

同じ音が複数の方向にある音源から同じ音量で耳に到達した場合に，もっとも速く耳に到達した音の音源方向に音源の定位が偏ってしまう現象。

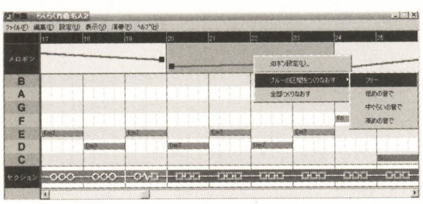
たとえば，リスナーから同じ距離にある左右二つのスピーカから同じ音量で同じ音を再生すると，音像は左右の中央に定位しているように聞こえます。しかし，そのどちらかのスピーカにほんの少しだけ時間差を与えると，時間差がないほうのスピーカ（左側のほうに時間差を与えた場合の，右側のほう）に音像が定位しているように聞こえてしまうのです。これを「ハース効果」と呼びます。

### カクテルパーティ効果

騒がしい場所（パーティ会場など）でも，特定の（聴こうと意識した）人の声や音を他の音と分離して聴きわけられる現象。当たり前のことだと思われるかもしれませんが，これは，聴覚の機能の中でも非常に強力な機能の一つで，その理由もいまだにはっきりとは解明されていません。

指定して編集するには，編集したいメロポン・アイコンをクリックして選択します。選択されたアイコンは色が変わります。

たとえば，サビ以外のメロディを作り直したい場合は，ブルーのメロポン・アイコンをクリックしてください。マウス右ボタンでクリックした後に，サブメニューの中から［ブルーの区間をつくりなおす］を選びます。さらに四つの条件がメニュー表示されますので，いずれか一つの条件を選びます。すると選んだ条件に合わせて，ブルーのメロポン・アイコン区間のメロディが作りなおされます。また，メロポン・グラフが自動的に変更されます。サビの部分のメロディを作りなおすときは，黄色のメロポン・アイコンを選択した後に，これと同様に選択していけば作りなおすことができます（**写真9**）。

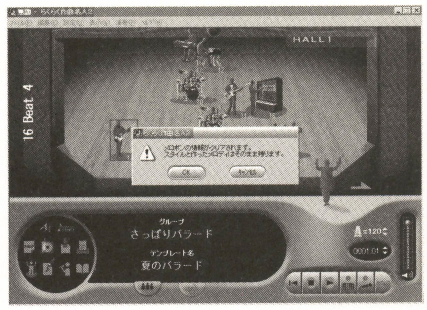


**《写真9》メロポン・アイコン区間のメロディの変更**

## アレンジを変える

メロポンボタンでメロディを作曲すると同時に，伴奏のアレンジもされてしまいますが，それまでのアレンジが気に入っているという人は，とくにアレンジを変える必要はないと思います。ただ，今まで3回分の連載で勉強してきたアレンジ方法を使って，よりイメージに合ったアレンジに変えてみてもいいでしょう。スタイルパレットでスタイルを変えたり，アレンジ画面でコードを変えたり，セクションを編集したり，ステージ画面でテンポを変えてみたり，音色を変えたり，ステージエフェクトパレットでリバーブのかかり具合を変えたり，いろいろ挑戦してみてください。

思い切ってアレンジボタンを使ってコードやスタイルをまったく新しく作り変えてみるのもおもしろいかもしれませんね。このときは，メロポンの情報がクリアされるので一度保存してからするといいと思います。一応アレンジボタンをクリックすると，このことが注意文で表示されますが，保存していたら［OK］をクリックでよいでしょう。保存していなかったら，オートアレンジする前の状態には戻れないので注意しましょう（**写真10**）。



**《写真10》メロポンの情報がクリアされるので，保存しよう**

さて，アレンジも終わったところで，らくらく作曲名人2による完全オリジナルの個性的な曲ができあがったんじゃないでしょうか？ホームページ上では作品発表も行っていますので，応募お待ちしています。

## その他の音符入力機能

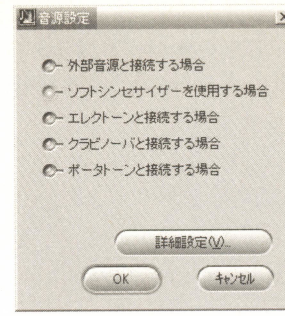
### 歌声録音

らくらく作曲名人2では，コンピュータに接続したマイクを使って，歌うことによるメロディパートの音符入力（MIDI入力）ができます。楽譜の読み書きができない人にとっては非常に便利な機能だと思います。また，思いついたアイデアを忘れないうちに保存しておくような，テープレコーダ的使いかたもできると思います。

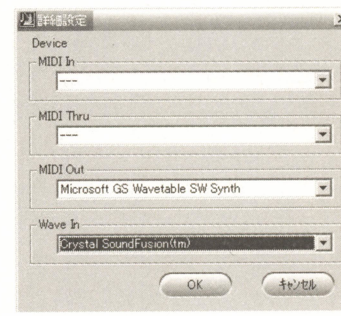
歌声録音をするためには，まずコンピュータ（あるいはサウンドカード）とマイクを接続する必要があります。コンピュータによって接続の方法が若干異なると思いますので，お手持ちのコンピュータの取扱説明書などを参考にしてください。ちなみに，僕のコンピュータの場合は，本体背面にマイク接続用端子を示すシールが貼ってあります。

つぎに，コンピュータが歌声を認識するかどうかの確認をします。［設定］メニューから［音源設定］を選ぶと，音源設定画面が開きます（**写真11**）。

音源設定画面の下側にある［詳細設定］ボタンをクリックすると詳細設定画面が開きます。画面の上から4段目［Wave In］で使っているサウンドカードに合ったデバイスドライバが正しく選ばれているかを確認します(**写真12**)。

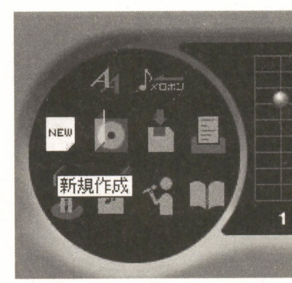



**《写真11》音源設定画面**




**《写真12》音源設定の詳細設定画面**

ここまでが完了したら，画面左下の［新規作成］ボタンをクリックし，新規ファイルを作ります（**写真13**）。




**《写真13》新規作成ボタン**

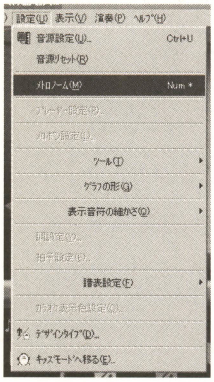
そして，下面中央下のコンダクター・ボタンをクリックし，コンダクターパレットを開きます。そこからいずれかの拍子ボタンをクリックし，これから録音する曲に合った拍子を選びます（**写真14**）。



**《写真14》コンダクターパレットを開き，拍子を選ぶ**

その後，最も録音しやすいテンポに設定（録音後に変更可能）し，録音しやすいようにメトロノームを鳴らすための設定をします。［設定］メニューから［メトロノーム］を選びます。これ

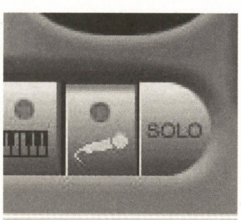
で，録音時にメトロノームがテンポに合わせて鳴り出すので，演奏のタイミングを取るガイドになります(**写真15**)。



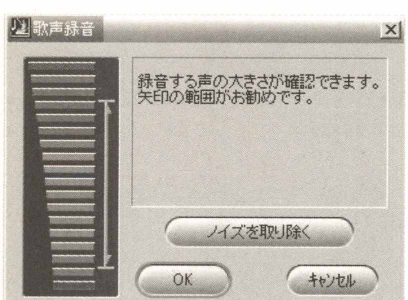

**《写真15》メトロノームを鳴らすための設定**

ここで一つ注意することがあります。メトロノームを鳴らしながら録音する場合に，スピーカを使ってモニタリングすると，歌声だけでなくメトロノームの音までもが録音されてしまいます。そこで，この場合のモニタリングはヘッドフォンを使って行ってください。

続いて，録音するパートを選びます。ステージ上のパート1のプレーヤーをクリックし，画面右下のマイクのイラストが付いた［歌声録音］ボタンをクリックします。歌声録音画面が開きます(**写真16，17**)。




**《写真16》［歌声録音］ボタン**




**《写真17》音量調整レベルメータ**

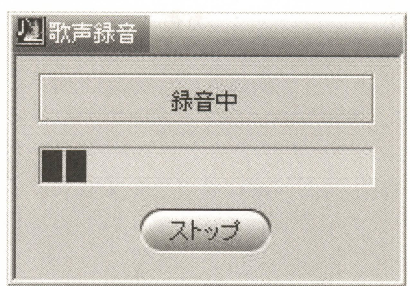
試しにマイクに向かって歌ってみて，音量がレベルメータ右の矢印の範囲内になるように声の大きさなどを調節します。ノイズが多い場合は［ノイズを取り除く］をクリックしてください。［OK］ボタンをクリックすると，自動的に楽譜画面に切り替わります。

つぎに，録音開始小節を設定します。［歌声録音］ボタンの上にある［演奏位置を進める］ボタンと［演奏位置を戻す］ボタンをクリックして調節します。あとでイントロなどをつける場合は，10小節くらい空けておくと便利です(**写真18**)。




**《写真18》録音開始小節を設定ボタン**

ここまでできたら，あとは録音するだけです。［再生］ボタンをクリックすると2小節のカウントが入った後，録音がはじまります。マイクに向かって歌うと（歯切れよく歌ってください)，歌った音が音符として表示されます。録音を止めるときは［ストップ］ボタンを押します。録音したデータを聴く場合は，［再生］ボタンをクリックします(**写真19**)。




**《写真19》録音中の画面**

録音をやり直すには，［編集］メニューから［元に戻す］を選んで録音を取り消します。録音開始位置などを再度調節し，［歌声録音］ボタンと［再生］ボタンをクリックして録音しなおしてください。

録音後に表示された音符は，普通に打ち込んだMIDIデータと変わらないので，音の高さや位置，長さなどを修正することができます。

## MIDIキーボードを使った録音

らくらく作曲名人では，MIDIキーボードを使って，リアルタイムでの音符入力（MIDI入力）が可能です。先ほどの歌声録音と同じ要領で録音できるので，前述の説明に沿って進めたいと思います。まず，キーボードとコンピュータ（あるいはサウンドカード）を正しく接続してください。ここでも，それぞれの機器によって接続の方法が若干異なるので，お手持ちの機器の取扱説明書を参考にしてください。

歌声録音でも説明した音源設定画面を開いて，使用する音源が正しく選ばれているか確認します。その後，詳細設定画面を開き，一番上の［MIDI In］でキーボードに合ったデバイスドライバが正しく選ばれているかを確認します。MIDIキーボードをコンピュータのシリアルポートに直接接続している場合は，「YAMAHA CBX Driver」を選んでください。

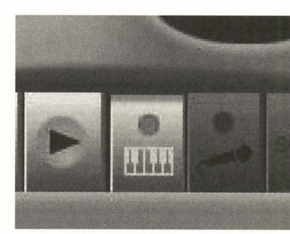
その後は，メトロノームを設定するところまで前項の歌声録音とまったく同じですので，前項の説明を参考にしてください。

設定完了後，録音するパートと楽器を選びます。ステージ画面で録音したいプレーヤーをクリックし，楽器パレットから楽器を選びます（**写真20**)。



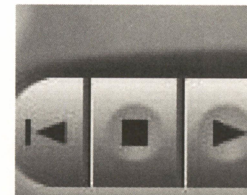

**《写真20》MIDIキーボードを使った録音は，設定完了後，録音するパートと楽器の選択**

鍵盤のイラストが付いた［MIDI録音］ボタンをクリックし，録音待機状態にします（**写真21**)。



**《写真21》［MIDI録音］ボタンで，録音待機状態**

そのまま録音開始小節を設定し，［再生］ボタンをクリックすると，2小節のカウントの後，録音がスタートします。録音をストップするには，［停止］ボタンをクリックします（**写真22**)。



**《写真22》録音をストップする［停止］ボタン**

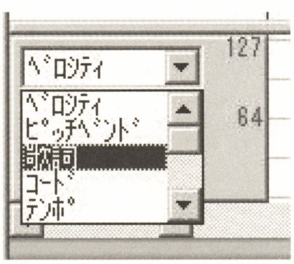
録音のやり直しも，前項と同じ方法で可能です。また，録音後の音符も歌声録音の場合と同じく修正することができます。

## 歌詞入力（カラオケ）機能

作った曲に歌詞データを入力しておけば，カラオケ画面で歌詞を表示させることができます。

歌詞の入力は楽譜画面，ピアノロール画面，ドラム画面のどの画面でも行えますが，ドラム画面やメロディ以外の楽譜を見ながらやる人はまずいないと思うので，パート1の楽譜画面で説明します。

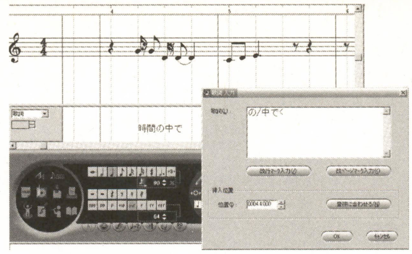
ステージ画面で，歌詞データを入力したい曲のファイルを開き，パート1のプレーヤーをダブルクリックします。表示された楽譜画面の左下のドロップダウンリストで，［歌詞］を選びます。歌詞は，その真横のコントロールセクションに入力していきます（**写真23**）。




**《写真23》カラオケ機能。楽譜画面左下のドロップダウンリストで，［歌詞］を選択**

画面中央下のツールボタンから鉛筆ツールを選び，歌詞を入力したい位置をクリックすることで歌詞入力画面が開きます。コントロールセクション下のスクロールバーを使って，音符の位置を確認しながら入力位置を選びます。選んだ位置に文字をタイプ入力します。こうした要領で歌詞を入力していきます。また，［改行マーク入力］ボタンや，［改ページボタン］を使って，カラオケ画面での改行，改ページの指定も行えます。改ページマークは次ページの歌詞の頭に付けてください。歌詞を入力したら［OK］をクリックします（**写真24**）。

歌詞の入力が終わったら［表示］メニューの［カラオケ画面］を選びます。この画面で曲を再生させると入力した歌詞の色がメロディに合わせて変わっていきます。これで，オリジナルのカラオケの出来上がりです。



**《写真24》歌詞の入力画面**

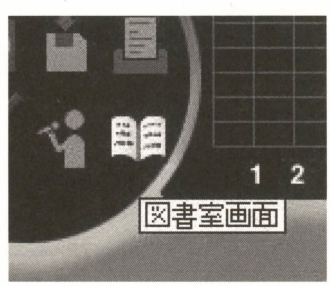
## 図書室機能

らくらく作曲名人には，使いかたのガイドや，さまざまなタイプの用語辞典を使用することができる図書室の機能があります。他のソフトにもあるような，ヘルプメニューとほぼ同じ役割を果たすと考えてもらっていいと思います。使いかたがわからないときなど，何か問題があるときはこれを使用するといいでしょう。

それでは，具体的な使いかたを説明していきます。

まず，図書室画面の開きかたについてですが，

- ［表示］メニューの［図書室画面］を選ぶ
- ［ヘルプ］メニューの［図書室へ］を選ぶ
- 画面左下のメニューの［図書室画面］ボタンをクリックする（**写真25**）

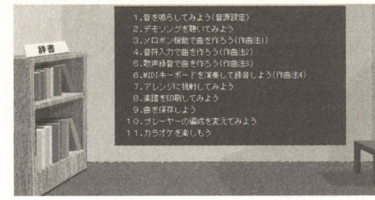



**《写真25》図書室画面ボタン**

以上三つの方法がありますが，どの方法でも開くことができます。また，図書室画面はどの画面からでも開くことができるので，わからないことが出てきたとき，すぐに調べることができます。

### ガイド

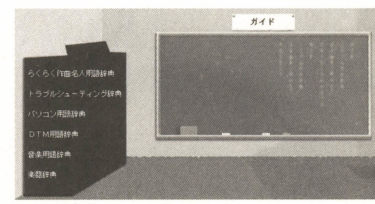
図書室画面の黒板の上にマウスポインタを移動するとガイドメニューが表示されます。そこで，見たいガイドの項目を選ぶと，選んだ項目のガイドが起動します（**写真26**）。




**《写真26》図書室画面の黒板の上にマウスポインタを移動するとガイドメニューが表示される**

### 辞書

図書室画面左側の本棚の上にマウスポインタを移動すると，辞典のメニューが表示されます。見たい辞典を選ぶと，選んだ辞典が起動します（**写真27**）。




### ◆Webの内容

- 芸大生の「らくらく作曲名人2」データ
- コンテストのくわしい応募方法

### ◆次回予告

- 自作マイクを作ろう（番外編）

## 「らくらく作曲名人2」作品募集！

前回募集したモチーフ（メロディ）をなんらかのジャンルにアレンジした曲の作品発表がホームページで行われています。「らくらく作曲名人2」で作った曲の作品発表会はまだまだ続きますので，どしどし応募してください。

今回の募集は，メロポンボタンを使って作曲した曲の募集です。前回まではアレンジを中心に募集してきましたが，今回は完全なオリジナル作品の募集ということで，前回までに勉強したアレンジ方法をうまく使って個性的な作品が期待できそうですね。また，今回説明した，歌詞入力機能を使って作詞までした作品も大歓迎です。作品の発表会は来月のWeb上で開催します。

参加希望の方は以下のデータを用意し，「らくらく作曲名人2・作品募集」係までEメールか，郵送でお送りください。

**・作品応募の際，用意するもの**

①モチーフアレンジ・データ，MIDI・xwsファイル
②作品の簡単な紹介，コメント
③作曲者プロフィール
④連絡先（住所，Eメール・アドレス）
（郵送の場合は，フロッピーディスクで）

http://www.basicmagazine.net/
http://www.beeplus.or.jp/bm/art/

大人気クリエイター集団「ギガ連射」があれこれ語る

# GIGA RENSYA つつうらら

**番外編**

**ギガ連射です。1か月の間をおいて，またベーマガに帰ってきました。「ギガ連射のヤロゥ！原稿落としやがったのかな？」とか「あれ？唐突に終了なの？」と思われた読者の方も多いと思いますが，すべては予定調和の世界であり，水戸黄門の印籠のごとく…。ゴホン，とにかく今月もお付き合いお願いします。**

## 『この１年を振り返る？』

えぇ〜っ，冒頭で「帰ってきたよ！」と意気込んだのは良いのですが，じつは今回で「ギガ連射　つつうらら」は本当に最終回なのです。残念ですが，お別れの時はいつか来るもの。あえて涙は見せますまい。

「ギガ連射　つつうらら」を連載してきたこの1年の間，結構ギガ連射も変わりました。なんと言ってもみなさんの応援あって，いろいろな活動の場が増えていったことですかねぇ。連載が始まった当時は，「ギガ連射？何それ？」って人が大多数で，ごく一部の人たちが「クターってアレだよね？」とマニアック振りを発揮していた時代でもありました。

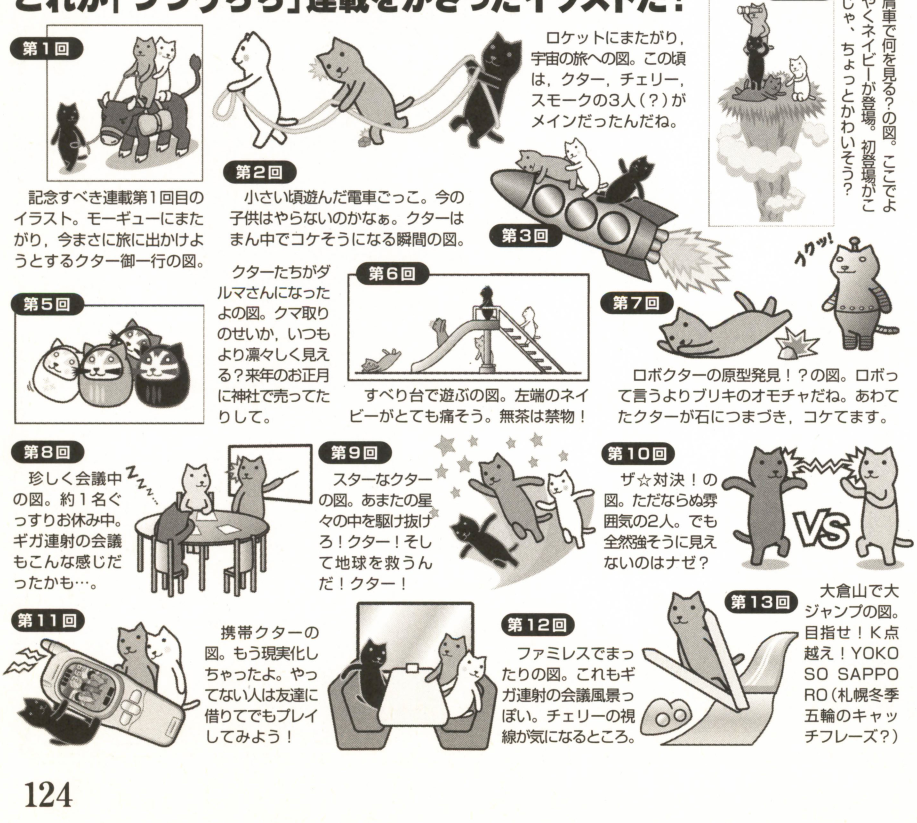
そのうち，「名づけてねっと」のイメージ・キャラクターに大抜擢されたり，あのJTBにも声をかけてもらったり，今度は携帯電話に進出ってんだから，まるで夢のような毎日でしたよ，えぇ。バージンロードを歩く花嫁の気持ちとでも言いましょうか（いやいや，ギガ連射は男所帯なんですがね）。

でも，変わらなかったこともあるんです。それは，「とにかく，おもしろいものを作っていこう！」ってこと。あ，今ちょっとだけカッコイイこと言ってる（自己陶酔）。もともと「ギガ連射　つつうらら」は，僕たちの制作の裏側を，オモシロおかしく紹介していこうというコンセプトのもとスタートしました。七転八倒しながらゲームを作ったり，たまには息抜きに大倉山ジャンプ競技場（前々月号参照）に行ってみたり，時にはダラけたりもしたけれど，そんな紆余曲折がクターのゲームのオモシロさにつながっていったんじゃないかなーって思います。

これから，何か「作っていこう！」って考えている人は，ギガ連射を真似してドタバタしなくてもいいですから，いろんなことに挑戦し，いろんな経験をつんで，もっともっとオモシロい物を作っていってほしいです。

とはいえ，僕らもここで終わったわけじゃないですから，これからもユーザーの期待をいい意味で「裏切り」つつ，爆進しますよ〜！

## これが「つつうらら」連載をかざったイラストだ！




第1回　記念すべき連載第1回目のイラスト。モーギューにまたがり，今まさに旅に出かけようとするクター御一行の図。

第2回　小さい頃遊んだ電車ごっこ。今の子供はやらないのかなぁ。クターはまん中でコケそうになる瞬間の図。

第3回　ロケットにまたがり，宇宙の旅への図。この頃は，クター，チェリー，スモークの3人（？）がメインだったんだね。

第4回　肩車で何を見る？の図。ここでようやくネイビーが登場。初登場がこれじゃ，ちょっとかわいそう？

第5回　クターたちがダルマさんになったよの図。クマ取りのせいか，いつもより凛々しく見える？来年のお正月に神社で売ってたりして。

第6回　すべり台で遊ぶの図。左端のネイビーがとても痛そう。無茶は禁物！

第7回　ロボクターの原型発見！？の図。ロボって言うよりブリキのオモチャだね。あわてたクターが石につまづき，コケてます。

第8回　珍しく会議中の図。約１名ぐっすりお休み中。ギガ連射の会議もこんな感じだったかも…。

第9回　スターなクターの図。あまたの星々の中を駆け抜けろ！クター！そして地球を救うんだ！クター！

第10回　ザ☆対決！の図。ただならぬ雰囲気の2人。でも全然強そうに見えないのはナゼ？

第11回　携帯クターの図。もう現実化しちゃったよ。やってない人は友達に借りてでもプレイしてみよう！

第12回　ファミレスでまったりの図。これもギガ連射の会議風景っぽい。チェリーの視線が気になるところ。

第13回　大倉山で大ジャンプの図。目指せ！K点越え！YOKOSO SAPPORO（札幌冬季五輪のキャッチフレーズ？）

## というわけで

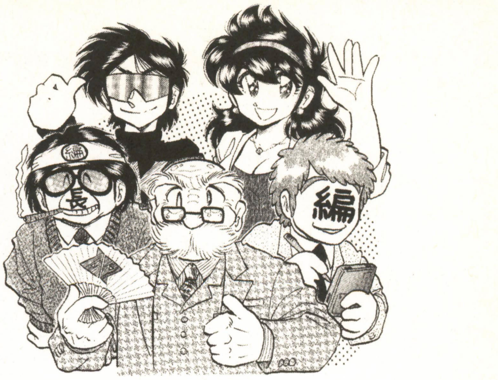
今まで応援してくださったベーマガ読者のみなさんに心ばかりのお礼です。

今まで「つつうらら」で登場してきたクターのイラストを壁紙にしました。

URLはhttp://www.giga-rensya.com/ba-maga/arigatou.html　です。ぜひ記念にダウンロードしてくださいね。

感想・質問・交際の申し込みなど，みなさんからのメールお待ちしています。
**otayori@giga-rensya.com**
**ギガ連射WEBサイト**
**http://www.giga-rensya.com/**

# あくせすROOM



## お便り連続回答

編：和歌山県橋本市・ヅラChuくんからのお便りに連続回答！

ヅ：最近は「○○さんのファンです」とか聞かなくなりましたね。

編：ご期待に応えて，載せてみました。

ヅ：君はまだ来ない，闘技場CD-ROM。

編：編集作業が思いのほか苦労しているようです。

ヅ：6月号の続報。あの宣伝を見てKousen Townに来てくださったかたがいました。感謝♥

編：どういたしまして。

---

編：宮城県仙台市・ぱぷわⅡ世くんからの連続回答！

影：なんでⅡ世なんだろう。

ぱ：別の雑誌で全然載らなかったので，ペンネームを変えて心機一転です。

編：ベーマガになにを望んでいるのかな？

ぱ：どんどん投稿して，載って，自分のコーナーを持つのが夢です。

つぐ美：文章投稿が減ってることについてコメントは？

ぱ：見直しもしていない作品を送るのも問題ありますが，内容的にはもっと素朴で，普通のテーマでかまわないと思います。

影：ふー，どうにかつながったぞ。回答だけ送ってくるってのもすごいなあ。

編：投稿はつねにほかの投稿者と編集者を出し抜くのが重要なんですよ。テーマ葉書でネタがカブらないのは当然として，募集していないものまで応募するチャレンジ精神が，新しい誌面を作っていくのです。

影：まあたしかに「こういう企画はどうですか」っつーより，「こういうの書いてみました」のほうが説得力あるからね。

Dr.D：ゲームの企画を書くにしても，ある程度プログラムを書ける人間は，具体的に説明ができるぶん，部活動やサークルの打ち合わせでも，プレゼンテーション面で有利に立ち回れるはずじゃな。

影：でも，こっちの手間が増える企画はちょっとなー。

Dr.D：たまには楽しいものじゃろう。

---

編：つぎは，よくある質問に連続回答。

つぐ美：電波新聞社には全号のベーマガが保管されてるんですか？

編：もちろん！　電波新聞社の地下は巨大な書庫になっていて，定期刊行物はもちろん，時代時代のさまざまな資料が整理されて保存されていますよ。

つぐ美：ベーマガの定期購読の送料って無料なんですか？

編：そうです。購読料は，消費税込みで6,120円です。かなりお得でしょう？

つぐ美：原稿料から引かれた源泉徴収を取り戻すにはどうすれば？

編：ベーマガにプログラムが掲載されると，これは原稿料扱いですので，小説や漫画と同じように，源泉徴収という形で1割の税金を先取りされてしまいます。課税の対象になるほど収入の多くない人は，税務署で確定申告をすれば，これは全額戻ってきますよ。いろいろと面倒かもしれませんが，社会勉強にもなりますので，ネタ作りのつもりで挑戦してみてはどうでしょう。2001年の原稿料で税金を取られ過ぎている人は，来年の2月16日から3月15日の確定申告で還付してもらいましょう。

## うろおぼえ友の会

「萌えプロ」につづく募集企画「うろおぼえ友の会」。それは，うろ覚えの妙，あるいはワザとボケてみる巧。資料を見ないで，お題をイラストに起こしてみてください。みなさんが学校で遊んだことのあるパーティ・ゲームを，誌上で楽しもうというコーナーです。応募期間を延長して，2001年11月15日必着でお送りください。

テーマ：原動機付き自転車

宛先：あくせすROOM
　　　うろおぼえ友の会係



◀埼玉県北足立郡・ぶんちょ
編：宛先が「でじこ特集うさだ係」になっていたので，なぜかこちらにまわってきました。
つぐ美：めざせ全キャラ掲載！




▶大阪府豊中市・藤井朱丸
編：インターネット上にウェブサイトを公開している人も多いですから、いろいろ探してみるのはどうでしょう。

## 素材名人 萌えプロ 最終回

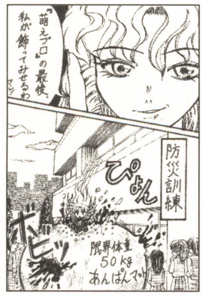
影：感動の最終回は，食欲とスポーツの秋ということで，防災訓練と菓子パンというコンビネーションで出題してみた。避難訓練にもいろいろあって，地震，火災，強盗，異常気象……いろいろなシチュエーションがあるわけで，ちょっと下調べをすれば，いろいろなアイデアが出てきたはず。集まってきたものは……萌えないなあ。出題失敗？



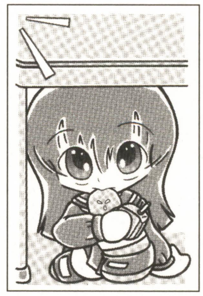
▲愛知県丹羽郡・裁刃鋭人
編：オーソドックスですが，怒りが伝わってきますね。でも，しっかりジャージ姿ってことは，訓練があることは知らされていたのでは？



◀大阪府松原市・鈴木真理子様江
影：なんか妖怪が楽しげに暴れているようだ。バケツとかのアイテムをもっと凝ればよかったね。




◀千葉県柏市・鈴木有人
つぐ美：飛び降りたときに勢いがついてるから、仕方ないのよー。仕方がないのよー。



◀愛知県刈谷市・りんす
編：こちらは口にくわえたままです。机の下の狭さの表現は，なかなかです。



◀福島市・みなみとらいば
影：「牛食え」ネタで投稿プログラムやイラストが載ると、抽選で記念品がもらえるという「牛食え報奨制度」が検討されています。

## あくせすROOM応募のルール

●愛読者カードや，官製ハガキ，同サイズの丈夫な紙などでお送りください。

**ワンポイント**
まとめ送りで大量に送るよりも，気合いの入った数枚の作品のほうが，インパクトは強く，掲載率は上がるようです。

●鉛筆や色鉛筆，涙滴の多いボールペン画は印刷に不向きなので避けましょう。

**ワンポイント**
下書きのあとが消し切れてないと見苦しくなり，評価が大幅にダウンしてしまいます。仕上げはていねいに。

●投稿内容は，掲載されたプログラムの感想や，学校でのできごと，一発ギャグなど，なんでも結構です。みなさんで楽しいページにしていきましょう。

≪応募先≫
〒141-8715 東京都品川区東五反田1-11-15
電波新聞社マイコンBASICマガジン編集部
あくせすROOM係

webでも受付中です。Internet読者ハガキでどうぞ。
http://www.basicmagazine.net/

SEGAのヘンな話題やちょっとお得な情報ならおまかせ!!

# ベーセガ

マイコン BASIC
SEGA OFFICIAL VARIETY SPACE

**奈美** ひゃっほー！　ゲームショウという一大イベントも無事終わって，元気いっぱい夢いっぱいのゲーマー奈美ですっ。…ってアレ，竹崎さんはどこへ？

**竹崎** あんさんは元気やのう……。ベーマガ読者のみんなは元気かぁい？　いてッ！

**奈美** わお！　竹崎さん，なんで床で寝てんですか？　思わず踏んじゃいましたよ（汗）。

**竹崎** それがもう1か月前にひいた風邪が治らず，すっかりへろへろなのじゃ。風邪は万病のもと。みんな気をつけような。

**奈美** さて，ゲームショウでどんなタイトルが発表になったか気になりますよねえ。

**竹崎** ほうほう，その話をこの病人にも聞かせてやってくださらんか。ゴホゴホ。

**奈美** いろんな発表があったんですけど「風来のシレン外伝」がDCで出るってことにびっくりした人も多かったんじゃないかな。

**竹崎** 1000回遊べるRPGで，DCユーザーに眠れない夜がやってくる，と（笑）。

**奈美** …で，寝不足によって竹崎さんの風邪がさらに悪化すると（笑）。あとは，「ファンタシースターオンライン」と「シェンムーII」がXboxで発売されるというニュースもありました！　今から楽しみですねー。

**竹崎** 大作とNo.1オンラインゲーム投入でXboxユーザーも寝かせない作戦なのじゃな。で，Xboxと言えばそのハードディスクやネットワーク機能を活かすべく「segaGT 2002」のアペンドディスクやオンライン版も発表になったよね。

**奈美** 新規タイトルがぞくぞく！

**竹崎** 他にも，DCでは「Jリーグ　スペクトラルサッカー（仮）」やDCとPS2で発売になる「スペースチャンネル5　Part2」。

**奈美** うららは，J-phoneでも活躍してますよね。

**竹崎** PS2では，人気シリーズの最新作「サカつく2002（仮）」「やきゅつく2002（仮）」が発表されたぞ。他にも「ハンドレッドソード」「カルドセプトセカンド」「ル・マン24アワーズ」「エアロダンシング4」。そしてそして～個人的には「SWITCH」の発売が何より嬉しいぜ～！

**奈美** 「SWITCH」って…その昔，メガCDで出たというソフトですよね？

**竹崎** こんなヘンなソフトは，その後1本もお目にかかってないくらいスゴイ内容だ。もう，絶対注目するしか……オエ～。

**奈美** な，なんですか，気持ち悪い！

**竹崎** いや，「SWITCH」の話をするにはこれが必要なんだよ。オエ～。

**奈美** 気持ち悪いから向こう行っててください！　PS2とNGCで「エターナルアルカディア」が発売されることも決まりました。

**竹崎** PSoneでは「シャカっとタンバリン」も。

**奈美** そしてGBAで発売される「シャイニングソウル」は，長年のシャイニングシリーズファンの方にはたまらないでしょうね。

**竹崎** いよいよセガもマルチプラットフォームに向けて本格始動してきた，って感じかな？　非セガファンだった人，買ってね。

**奈美** そして，11月22日には，いよいよPS2参入第1弾となる「Rez」が発売になります！

**竹崎** 「Rez」と言えば，気になってはいるんだけど，イマイチ内容がよくわからない，っていう人もいそうだよね。

**奈美** ですね～。実際にやってみると一目瞭然なんですけどね～。

**竹崎** 基本操作は，敵をロックオンして撃つ，ってゆー，ただそれだけなんだけど…。

**奈美** それだけなんだけど，それがだんだん気持ち良くなってくるんです，本当に！

**竹崎** シューティングゲームとかでも，敵を一掃できると気持ち良かったりするよね？

**奈美** でも，「Rez」の場合は，それに加えて，音と光と振動という要素があるんですよ。

**竹崎** 同じように敵をやっつける場合でも，1匹のときと，4匹を一度にやっつけるときとでは音や光が変化するんだよね。

**奈美** つまり，こちらが撃つと音が返ってくる。「撃つ」「返ってくる」「撃つ」「返ってくる」。これを繰り返すことによって高揚感というか，ナンダカ気持ち良くなってきちゃうんです。

**竹崎** さらに，振動もこれに加わるもんね。

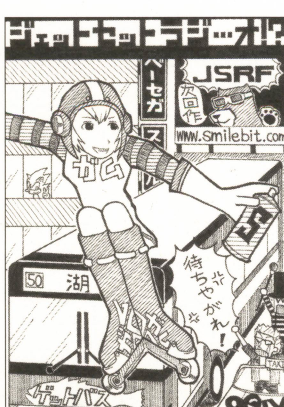
**奈美** もー振動は絶対に外せませんね。DC版なら，ぷるぷるぱっく，PS2版ならデュアルショックに加えて，トランスバイブレーター（別売もしくは初回同梱版をお求めくださいね♪）は必須ですっ。さらに気持ち良くなれちゃうこと請け合いですよ！

**竹崎** 新しいエンターテイメントはセガから！ 秋の夜長は，「Rez」でウッボー（笑）。



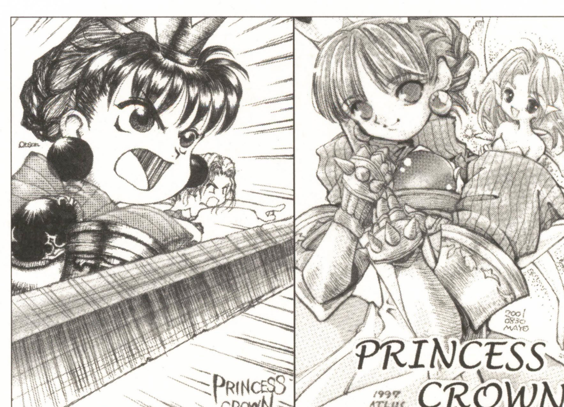

◀ぱんだぱぱ

そして今月もまたおたより募集ピョ！ ラッピー同盟隊員1号・ゲーマー奈美（笑）




◀バグニンゲン

ゲット「バス」の行き先は湖なんですね！ 他にもいろいろと楽しいイラストなんですよ。細かいトコも見てみましょ。




◀引越しおーすん＆たまごまよ

昔懐かしい（？）常連さんからのおたよりですっ。しかもご夫婦での投稿ですよ。ありがとうございますっ。なんだか盆と正月がいっぺんに来たみたいですね！ご心配をおかけしてますが、小熊さんにもよろしくお伝えください（笑）

このコーナーを充実させるための，ご意見，お便り，イラストなどなどクビを長くしてお待ちしています。採用の方には特製テレカなど，非売品セガグッズをプレゼント。SEGA竹崎がお便りをすべて読んで，面白いコーナーを作っていきたいと思います。すべてのあてさきは，ここや!!

〒144-8531　東京都大田区羽田1-2-12　（株）セガ　カスタマーゲートウェイ部「竹崎と奈美のベーセガ」係

セガ竹崎の「セガスタ」はこちら→http://www.sega.co.jp/segasuta/

# Falcom® NEWS EXPRESS

日本ファルコム株式会社
〒190-0012 東京都立川市曙町1-14-13
TEL 042-527-6501
FAX 042-528-2714

スタッフ募集中
採用担当BFN係

http://www.falcom.co.jp/




## 「Zwei!!（ツヴァイ）」サントラとイース完全新作が無料でもらえる！
## 初回限定特典版は完売必至！予約はお早めに！

**初回限定三大特典**

【特典1】オリジナル・サウンドトラックZwei!!（Audio CD）
【特典2】イース・ヒーリング（Audio CD）
【特典3】永久保存版「Zwei!!」特製シルバーメタリックBOX

「Zwei!!（ツヴァイ）」
12月20日発売
Windows95/98/Me/2000/XP

ファルコムの作品は毎回豪華な特典も話題となっていますが，Zwei!!の特典は，なんと12月20日に同時発売予定のFMS第5弾！まず特典1のサントラZwei!!は，イースや英雄伝説のサウンドとは一味違ったファルコムサウンドチームjdkの新境地とも言える，ツヴァイ!!の全曲集です。特典2のイース・ヒーリングは，あの熱く激しい「交響曲21stセンチュリー」とは趣の違う，名作プレプリマを彷彿とさせる“癒やし系”の完全新作新アレンジです。これだけでもかなり“お得”な特典版ですが，さらにYsⅠ・Ⅱ完全版・ゴールドケース入りイースの書パッケージ」を越えるべく鋭意制作中の特製シルバーメタリックBOXとなっています。初回限定ですので予約だけで完売してしまう可能性が非常に高いため，ご予約はお早めに！

ご予約は簡単便利なオンラインショッピングコーナーをぜひご利用ください。

http://www.falcom.co.jp./zwei/order.html

## スタッフのこだわりと遊び心が満載！ファルコム新作「zwei!!（ツヴァイ）」に乞うご期待！

遂に発売日も12月20日に決定し，ますます盛り上がってきている，ファルコムの完全新作「ツヴァイ!!」は，これまでのどのシリーズにも属していない全くのオリジナル新作タイトルです。主人公は，遠距離攻撃のピピロ（魔法使い）接近戦が得意なポックル（戦士）の2人ですが，このコンビ，実にいい味だしています。ジュリオ＆クリス（白き魔女）のコンビに負けてません。で，この2人をうまく使い分けながら冒険を進めていくアクションPRGですが，自由度が極めて高く，どのステージから攻略していくかはプレイヤー次第。もちろんダンジョンの最奥部にはボスが控えているわけですが，画面に収まりきらないほどの巨大なボスが続々登場します。さらに本筋には関係ないストーリーが用意されていたり，シューティングゲーム「エアロぴぴろ21stセンチュリー（仮）」などもあったりと，スタッフのこだわりと遊び心が満載です。ファルコムが贈る今年の年末最大の話題作［zwei!!」に乞うご期待！

## 「zwei!!」攻略本の発売も決定！

アクションはちょっと苦手と思っている方に朗報！角川書店から「Zwei!!」の攻略ガイドの発売が決定！これで確実にエンディングを迎えることができるぞ！現在，「Zwei!!」と同時発売を目指して制作中とのこと。単なる攻略だけでなく，冒険に役立ついろんな情報をすべて網羅し，読み物としても楽しめる構成になるそうなので，攻略本がなくてもクリアできる！と思っている方でも，ぜひ読んでZwei!!の世界を堪能しよう！

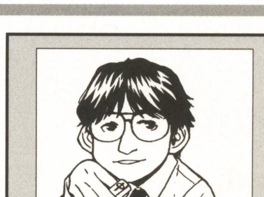
## PCゲーム定番キャンペーン実施中！

超便利な極薄最新型マウスシートがもらえるファルコム定番キャンペーン。スタートから約1か月経ってしまいましたが，早くもマウスシートがなくなったお店も続々。期間は11月30日までだけど，マウスシートがなくなった時点で終了となるため，狙っている方はお早めに！キャンペーン実施店は，ファルコムホームページのショップリストをチェック！

http://www.falcom.co.jp/shoplist/index.html

## ファルコムのイラスト集が12月に発売決定！

これまでありそうでなかった，ファルコムのイラスト集の発売が決定したぞ！ファルコム20年間の活動の中で描かれてきた数百点にも及ぶ中から厳選された永久保存版のイラスト集だ。最新作「Zwei!!」や未公開秘蔵イラストも収録予定ですのでお楽しみに！



先日FMS第4弾が発売されたばかりですが早くも第5弾が登場！しかも今回はZwei!!の特典版を購入された方は無料！特典版はすぐに完売する可能性がありますので，予約はお早めに！




タイトル冠文：大阪府・御堂筋太郎
タイトル・イラスト：大阪府・藤井朱丸
本文：暁

# 夢に出てくる なにわ通信

PRESENTED BY マイコンソフト

http://www.micomsoft.co.jp/

ADDRESS● 〒530-0005大阪市北区中之島3-2-4
朝日新聞ビル6F 電波新聞社大阪本社内
マイコンソフト株式会社　なにわ通信係

ちょっと気が早いですが，年末年始のボーナスやお年玉の使い道は，もう決まっていますか？

まだお金の使い道が決まっていない方には，マイコンソフトの新製品「XRGB-2plus」がおススメです。XRGB-2plusは，家庭用ゲーム機などの映像を，パソコン用ディスプレイで表示するための変換装置です。

## 発売日が若干変更に

新製品の「XRGB-2plus」は，発売予定が若干遅れそうです。

発売は，11月下旬に変更されました。

誠に申し訳ありません。

より良い製品を開発するために，若干時間が必要とのことです。

でも，クリスマスにはきっとみなさんの手元で活用されていると思います。

先日，幕張で行われた展示会「シーテック（旧エレクトロニクスショー）」でXRGB-2plusが展示されました。

遊びに来てくれたお客さんの反応は予想以上でした。

XRGB-2plusは，現行機XRGB-2を使用されている方も，「購入したい」と思うような製品に仕上がっております。

きっと皆さんを満足させてくれると思いますので，是非ご検討ください。

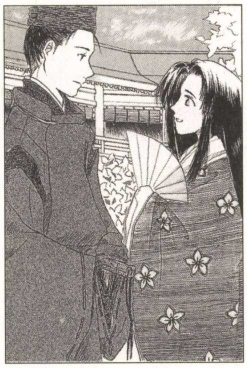
## 来年の新製品

マイコンソフトでは，チューナー内蔵型のアップ・スキャンコンバーターも開発検討中です。

こちらの新製品は，プログレッシブ表示が可能で，テレビ映像やゲーム機のインターレス画面（ちらつきのある画面）を，ノンインターレス化し，ちらつきのきわめて少ない映像方式（プログレッシブ）に変換することができます。この製品は，ビデオ入力端子とS入力端子を備えています。（RGB入力端子やD入力端子はありません。）

発売は，来年春です。ご期待ください。

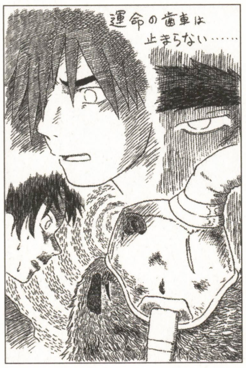
## ありがたいおハガキのコーナー



熊本県・香菜薫
北海道・春陽
おぉっ！香菜薫さんと春陽さんの合作です。香菜薫さん，大学入学おめでとうございます。



埼玉県・ぶんちょ
モジモジしている女の子です。近くに好きな人でもいるのでしょうか？視線がとてもいい感じです。




大阪府・ARAI
熊さん，いったいどうしたのですかーっ！熊さん危うし！どうしたらいいのか！主人公よ！今こそ過去を拭い去るときだ！



千葉県・某女史へ。大変失礼（笑）
きゃ～可愛い～！こんな鳥さんに乗ってみたいです～ でも食べてるゴキブリは，ちょっと気持ち悪いかも。（笑）



大阪府・茶鈴様江
題名は「村はずれにて」とのこと。旅のマジシャンと，村の女の子。猫のお人形を一生懸命作ったのでしょうね。



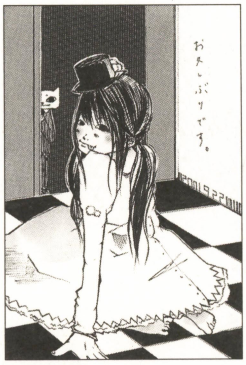
山形県・MAYO
お久しぶりです。あいかわらず素敵なイラストを描かれますね。天使の矢で私の胸を貫いてやってくださいませ。




山形県・おーすん
はじめまして！ですよね？大爆笑です。なにわ通信は狙われているのですね！新人君は裸のようですが，お尻がとても可愛い。



北海道・春陽
珍しいアングルですねぇ。たばこを吸う男の後ろ姿というのも，なかなか格好いいものですね。春陽さんお薦めの歌手はIVY。



北海道・青歌
修学旅行はどうでしたか？良い思い出を沢山作れましたでしょうか？受験勉強も頑張ってくださいね。



愛知県・邪道坊や
戦艦のオペレーター三人娘とのこと。みんなの憧れの的。いわゆるマドンナですね。

Q10　携帯・PHSを持っていますか？
①両方持っている ②携帯 ③PHS ④購入予定 ⑤予定なし

Q11　Q10で④の方のみ，どちらを購入する予定ですか？
①携帯 ②PHS ③両方 ④未定

Q12　使用中のプログラミング言語は？
※枠内に一つご記入ください

Q13　これから勉強したいプログラミング言語は？
※枠内に一つご記入ください

Q14　今月号の本誌で面白かった記事を三つ，下の表からお答えください

Q15　今月号の本誌でつまらなかった記事を三つ，下の表からお答えください

ご協力ありがとうございました。

## ■今月号の記事一覧



キリトリ線

**マイコンBASIC Magazine**　プログラム投稿応募カード

| | | | | | |
|---|---|---|---|---|---|
| 言語名（音源名） | | | | | |
| プログラム名（曲名） | | | | | |
| 氏　名 | | ペンネーム | | 年　齢 | 歳 |
| 住　所 | 〒 | | | | |
| TEL | ※連絡可能な時間帯 | | | 応募年月日 | 年　月　日 |

キリトリ線

郵便はがき

| 50円切手をお貼り下さい |

1 4 1 - 8 7 1 5

（受取人）

東京都品川区東五反田1−11−15

**電波新聞社**

**月刊 マイコンBASICマガジン 編集部**

12月号 行

| | | | |
|---|---|---|---|
| （フリガナ） | | 年 齢 | 性 別 |
| ご氏名 | | 歳 | 男 ・ 女 |
| □□□-□□□□ | | 電　話 | |
| ご住所 | 都・道 府・県 | | |

**職業**：小学生／中学生／高校生／高専生／短大生／大学生／専門学校生／予備校生／アルバイト／公務員／会社員／その他（　　　　）

**本誌を** 毎月・時々・初めて 読む　**何人で読みますか**（　　）人

**自分のホームページを**　持っている・持つ予定・持っていない・予定なし

**本誌のほかによく読む雑誌は？**［　　　　　　　　］

| | | |
|---|---|---|
| **ベーマガお楽しみプレゼント⑫ 応募欄（86ページ欄外）** | **希望番号** | **品名** |
| | 番 | |

キリトリ線

# 月刊マイコンBASICマガジン⑫ 愛読者カード

★OFコーナー宛 ♥あくせすROOM宛 ■質問・感想（本誌への掲載／可・不可）

ペンネーム：

## ■アンケート回答記入欄■

※黒の筆記用具でわかりやすく，はっきりとご記入ください

| | | | |
|---|---|---|---|
| Q 1 □ | Q 2 □ | 〔メーカー名 〕 | |
| Q 3 □ | Q 4 □ | Q 5 □ | |
| Q 6 □□□□□□□□ | | | |
| Q 7 □□□□□ | 〔商品名 〕 | | |
| Q 8 □□□□□□□□ | | | |
| Q 9 □□□□□□□□ | | | |
| Q10 □ | Q11 □ | | |
| Q12〔 〕 | Q13〔 〕 | | |
| Q14 □□□ | Q15 □□□ | | |

キリトリ線

## ■アンケートにご協力ください。

※左の愛読者カード・記入欄に，以下の質問に該当する番号をご記入ください。複数回答の質問で，回答欄をすべて埋める必要はありません。

Q1　メインで使っているパソコンは？
①デスクトップ ②ノート ③持っていない

Q2　パソコンの種類は？（Q1で③以外の方）
①自作 ②ショップブランド ③メーカー製
※枠内にメーカー名も記入してください

Q3　メインで使っているOSは？（Q1で③以外の方）
①Windows 95 ②Windows 98 ③Windows Me ④Windows 2000 ⑤Windows NT ⑥Macintosh ⑦UNIX ⑧Linux ⑨その他

Q4　パソコン歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q5　インターネット歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q6　パソコンの主な使用目的は？（複数回答可）
①プログラミング ②ゲーム ③インターネット ④メール ⑤ミュージック ⑥学習 ⑦CG ⑧その他

Q7　これから買いたいと思うコンピュータ機器は？（複数回答可）
①パソコン本体 ②プリンタ ③スキャナ ④デジタルカメラ ⑤その他
※わかれば，商品名も記入してください

Q8　持っているゲーム機は？（複数回答可）
①プレイステーション（PSone含）②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥ゲームボーイ ⑦ゲームボーイカラー ⑧ゲームボーイアドバンス ⑨ワンダースワン ⑩ワンダースワンカラー ⑪その他

Q9　これから買いたいと思うゲーム機は？（複数回答可）
①プレイステーション（PSone含）②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥Xbox ⑦ゲームボーイカラー ⑧ワンダースワンカラー ⑨ゲームボーイアドバンス

※裏面へつづく→

## ■メモ欄…プログラム担当者・編集部へのご意見・ご感想など



ファルコム完全新作オリジナルアクションRPG！
Windows 95/98/Me/2000/XP
ツヴァイ!!
Zwei!!
12月20日
発売!!
■2人＋1匹で大冒険！
ニャ！
ワン！
・近距離ポックル。遠距離ピピロ。
いつでも瞬時に切り替えて恐いものなし！
・ただのペットと侮る無かれ。
ご主人様をおたすけするニャ～！
アクションRPGのファルコムから！
イース、ザナドゥ、ブランディッシュ…etc。
数々の名作を輩出してきたファルコムの
アクションRPG史に、
今、新たなる歴史が
刻まれる———。
CRITICAL!
・リズムが重要？クリティカル！
タイミングを合わせて大打撃！
・属性を見極めろ！反属性なら大ダメージ、
同属性だと回復しちゃう～?!
■体当りだけじゃツマンナイ！
おまけの一撃!
・連続ヒットで必殺技獲得！
ヒットを貯めて超必殺技をお見舞いしよう。
■舞台は山あり谷あり、
空中あり!?
・一見ほのぼの。だけどダンジョンには危険がいっぱい！
きゃ～！
■謎、罠、隠し×××…。
・あなた好みの名前をつけてね。
エンディングを迎えた後もお楽しみいっぱい！
・ファルコムシューティング?!「エアロぴぴろ21stセンチュリー(仮)」
初回限定 豪華三大特典付！予約受付中！
——Zwei!!サントラとイース完全新作が無料でもらえる！——
【特典1】オリジナル・サウンドトラックZwei!! (Audio CD)
…イースでもない、英雄伝説でもない、ファルコムサウンド新境地！ツヴァイゲームサントラ全曲完全収録！
【特典2】イース・ヒーリング (Audio CD)
…永遠の旋律達に包まれて揺蕩う忘我の時。イース癒し系完全新作新アレンジ！
【特典3】永久保存版「Zwei!!」特製シルバーメタリックBOX
…自らに挑戦！「Ys完全版・ゴールドケース入りイースの書パッケージ」を超えるか？乞うご期待！
※「オリジナル・サントラZwei!!」と「イース・ヒーリング」の別売もあります。(12/20同時発売)
■隅々まで歩いて隠された
イベントを探せ!
■ファルコムゲーム史上
最大クラスのボスが続々!
「ツヴァイ!!」の最新情報は、ファルコムホームページをご覧ください。
http://www.falcom.com/
イラスト:七瀬 葵
PCゲーム 定番
PCゲーム定番キャンペーン(9/29～11/30)
キャンペーン期間中に、イースⅠ・Ⅱ完全版他、ファルコム製品をキャンペーン実施店で購入された方に、薄くて軽い超便利な最新型マウスシートを全員プレゼント！
詳しくは店頭に貼られてあるポスターをご覧ください。
キャンペーン実施店は、ファルコムホームページのショップリストに掲載中！お近くの方は要チェック！
キャンペーン参加店様募集中!
お気軽にお問い合わせください。
Falcom®
NIHON FALCOM CORPORATION
〒190-0012 東京都立川市曙町1-14-13
☎042(527)6501 http://www.falcom.co.jp/
■お求めは取扱い店またはカンタン便利なファルコム通販(falcom.comファルコムドットコム)もご利用下さい。







定価550円 本体524円 雑誌18361-12 T1118361120554

第二十巻第十二号通巻二三四号 昭和五十七年十二月六日第三種郵便物認可 二〇〇一年十一月二十五日印刷 二〇〇一年十二月一日発行

電波新聞社 〒141-8715 東京都品川区東五反田一－十一－十五 電話〇三（三四四五）六一一一

2001

わかる！動かせる！
プログラムが組める雑誌

# マイコンBASIC Magazine

12

特集 Windows XP導入実践講座

http://www.basicmagazine.net/

電波新聞社